

ERDC/CERL CR-06-1

Construction Engineering  
Research Laboratory



**US Army Corps  
of Engineers®**

Engineer Research and  
Development Center

# Absorption of Low-frequency Sound Energy by Vegetation

## A Laboratory Investigation

Ryan Joseph Lee

February 2006



# **Absorption of Low-frequency Sound Energy by Vegetation**

## **A Laboratory Investigation**

Ryan Joseph Lee

*Construction Engineering Research Laboratory  
PO Box 9005  
Champaign, IL 61826-9005*

through

*Department of Electrical and Computer Engineering  
University of Illinois  
Urbana, IL 61801*

Final Report

Approved for public release; distribution is unlimited.

Prepared for     U.S. Army Corps of Engineers  
Washington, DC 20314-1000

Under             Work Unit #3JD946

**ABSTRACT:** Many field research projects have been conducted to study the effects of natural foliage on the propagation and attenuation of sound. This research takes natural foliage into a controlled laboratory setting to test its low-frequency acoustic characteristics. Absorption of low-frequency components of unwanted noise is of interest to the Army, but has been an unsolved problem due in part to the cumbersome and expensive testing facilities needed to study long wavelengths.

In this research, low-frequency absorption and reflection coefficients were found reliably and consistently. Due to study of the steady state conditions, the methods presented here could constitute a more consistent method than ever before. The procedures described in this paper can serve as a handbook for future research; recommendations are included.

**DISCLAIMER:** The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products. All product names and trademarks cited are the property of their respective owners. The findings of this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

**DESTROY THIS REPORT WHEN IT IS NO LONGER NEEDED. DO NOT RETURN IT TO THE ORIGINATOR.**

# Contents

List of Figures and Tables .....	v
Preface.....	vii
Acknowledgements .....	viii
List of Symbols .....	xi
<b>1 Introduction .....</b>	<b>1</b>
<b>2 Testing Facility and Operation .....</b>	<b>3</b>
2.1 Impedance Chamber .....	3
2.2 Yokogawa ScopeCorder .....	5
2.3 Theoretical Operational Rationale .....	6
<b>3 <math>Q</math> (Quality Factor) .....</b>	<b>9</b>
3.1 Methods of Computing $Q$ .....	9
3.1.1 Method 1: the 3 dB-down method .....	9
3.1.2 Method 2: the energy method .....	10
3.1.3 Method 3: the logarithmic decrement method .....	11
3.2 Choosing a Method .....	13
3.3 $Q$ and $\alpha$ , Absorption Coefficient .....	16
3.4 Laboratory Techniques .....	16
<b>4 Quality Factor Results .....</b>	<b>19</b>
4.1 Empty Chamber .....	19
4.2 Vegetation in Chamber .....	20
4.2.1 Apple tree branches .....	21
4.2.2 Bales of straw .....	21
4.2.3 Northern white pine tree .....	23
4.3 Testing Limestone Crushed Rock .....	26
<b>5 Load Impedance and Reflection Coefficient .....</b>	<b>31</b>

<b>6</b>	<b>Standing Wave Results .....</b>	<b>38</b>
6.1	Vegetation Tests.....	38
6.2	Gamma Results.....	40
<b>7</b>	<b>Chamber Handbook .....</b>	<b>42</b>
7.1	White Noise $Q$ .....	42
7.2	$\Gamma$ Computation and Standing Wave Fitter.....	47
<b>8</b>	<b>Conclusion.....</b>	<b>55</b>
	<b>Appendix A: Theoretical Standing Wave Viewer .....</b>	<b>57</b>
	<b>Appendix B: Code to Find <math>Q</math> of the Chamber .....</b>	<b>59</b>
	<b>Appendix C: Code to Find Best Fit Standing Wave .....</b>	<b>65</b>
	<b>References .....</b>	<b>69</b>
	<b>Report Documentation Page.....</b>	<b>71</b>

# List of Figures and Tables

## Figures

2.1 Reinforced concrete impedance chamber with steel door. ....	3
2.2 Loudspeaker installed in the center of the back of the chamber.....	4
2.3 The microphone port on the left has a cable running out of it, whereas the other microphone ports are plugged. ....	5
2.4 A single frame of the output of RUNCHAMBER.....	8
3.1 A parallel resonant circuit .....	9
3.2 The bandwidth of a parallel RLC circuit whose resonant frequency is 20 Hz and where $Q = 1$ .....	10
3.3 This theoretical device has a steady response to a 20 Hz pure tone signal until the input is completely cut off at time $t = 0$ s. ....	12
3.4 The power spectrum of the input white noise is flat in the frequency range of interest, i.e., 15-100 Hz. ....	15
3.5 The magnitude of the DFT of a microphone in the chamber when white noise is the chamber's input.....	15
3.6 Standing current and pressure waves shown on a hypothetical transmission line with characteristic impedance $Z_0$ (a). The placement of even an infinitely large impedance at a current null does not affect the standing waves (b).....	17
3.7 Standing pressure waves in an empty chamber for the first four modes. ....	18
4.1 The transmission line equivalent of the empty chamber (a) and the chamber with some vegetation placed at some location along its axis (b).....	19
4.2 The microphones in the chamber measure pressure ( $V_{mic}$ in our analogy) at different locations. ....	20
4.3 For each resonant mode, the $Q$ of the chamber decreases with increasing vegetation amounts placed inside.....	22
4.4 Two branches (a) cut from another tree were used for extra vegetation mass and the top 130 lb. of the tree (b) at the chamber door.....	24
4.5 Indeed, the $Q$ is a function of the mass of the vegetation in the chamber. ....	25
4.6 A 2 m x 2 m two-level box. ....	26
4.7 The lower level of the box placed at the door side of the chamber.....	27
4.8 The full box placed at the door side of the chamber. ....	27
4.9 The lower half of the box filled with gravel. ....	29
4.10 The full box, fully loaded. ....	29

4.11 Eight bales of straw on top of the full box filled with limestone. ....	29
4.12 The empty chamber compared to the five other tests with varying amounts of rock and the box.....	30
5.1 The terminating load impedance is most likely complex.....	31
6.1 The chamber terminated with (a) 10-bale straw wall and (b) pine tree branches hung on the inside of the chamber door.....	38
6.2 Best fit standing wave patterns for a pure tone input at 70 Hz in a chamber terminated with different amounts of vegetation... ..	39
6.3 $ \bar{\Gamma}_r $ as a function of frequency for three chamber terminations: empty, pine tree sections, and bales of straw. ....	41
7.1 Ten bales of straw placed to terminate the empty chamber.....	47
7.2 The code outputs the best fit standing pressure wave as well as the data given at the top of the graphic: tone frequency, reflection coefficient, standing wave ratio (SWR), and load impedance. ....	48

## Tables

2.1 The acoustic/electric/mechanical analogy .....	7
4.1 Empty chamber $Q$ .....	20
4.2 Chamber $Q$ values with varying amounts of straw bales .....	23
4.3 Chamber $Q$ values with varying amounts of pine tree.....	26
4.4 Limestone properties.....	28
4.5 Limestone gradation.....	28
4.6 Chamber $Q$ values from rock experiments.....	30
6.1 Reflection coefficients .....	39

## Preface

The work was monitored by the Ecological Processes Branch (CN-N) of the Installations Division (CN), Construction Engineering Research Laboratory (CERL). The CERL Principal Investigator overseeing the work was Michael White. This work was done by Ryan J. Lee. Alan B. Anderson is Branch Chief, CN-N, and John T. Bandy is Chief, CN. The associated Technical Director was William D. Severinghaus. The Acting Director of CERL is Dr. Ilker R. Adiguzel.

CERL is an element of the U.S. Army Engineer Research and Development Center (ERDC), U.S. Army Corps of Engineers. The Commander and Executive Director of ERDC is COL James R. Rowan, and the Director of ERDC is Dr. James R. Houston.



## **ACKNOWLEDGMENTS**

This work would not have been possible without the support, guidance, and faith of Dr. George W. Swenson, Jr., and the acoustics team at the U.S. Army CERL. Specifically, Dr. Swenson assisted in my coursework, believed in my adaptability as a civil engineer transitioning into an electrical engineering curriculum, and came up with the idea for many experiments. His extensive knowledge is unsurpassed in our field and has challenged me to dig as deeply as possible.

Of the CERL acoustics team, Dr. Larry Pater provided funding as well as a wide variety of projects to give my research breadth. Dr. Michael White and Dr. Michelle Swearingen offered countless suggestions and brainstorming to help me achieve a more extensive level of understanding in acoustics. Jeff Mifflin assisted in technical expertise, equipment procurement, operations, and helped bridge the gap between theory and empirical results. Dr. Steven J. Franke offered his time, patience, and possible solutions at an important time in completing the thesis. My most sincere gratitude goes out to these friends and colleagues without whom this work would not have been executed.

## LIST OF SYMBOLS

$A$	amplitude
$\alpha$	absorption coefficient
$\beta$	$\omega/c$
$c$	speed of sound
$C$	capacitance
$f$	frequency
$\gamma$	specific weight
$\Gamma$	reflection coefficient
$\gamma$	attenuation constant
$k$	specific heat ratio
$I$	current
$l$	length
$L$	inductance
$\omega$	radian frequency
$p, P$	instantaneous sound pressure
$P^+$	positive-traveling pressure wave
$P^-$	negative-traveling pressure wave
$q$	charge
$Q$	quality factor
$R$	gas constant
$\rho$	instantaneous density
$\rho_0$	static density
$S$	cross-sectional area
$T$	temperature
$t$	time
$\theta$	phase angle
$U$	volume velocity
$V$	voltage
$Z$	impedance

## 1 INTRODUCTION

A multitude of research projects have been conducted to study the effects of natural foliage on the propagation and attenuation of sound. Scores of species have been tested and compared.<sup>1</sup> Other variables such as wind, temperature, humidity, and elevation have, for the most part, been out of the control of the researcher. This paper takes natural foliage into a controlled laboratory setting to test its low-frequency acoustic characteristics.

Three mechanisms that have commonly been studied and quantified are ground reflection, scattering, and absorption of sound. Absorption of low-frequency components of unwanted noise is of interest to the U.S. Army Corps of Engineers, but is an unsolved problem due to the cumbersome and expensive testing facilities needed for long wavelengths and parameters that are out of the control of the researcher.

Until now, low frequencies of sound (20–80 Hz) have not been studied in controlled laboratory conditions with testing apparatuses the size of a half wavelength or more. Until Burns<sup>2</sup> brought pine tree components into the laboratory, experiments were done in the field under nonideal conditions. However, his testing apparatus was a box much smaller than the half-wavelengths of low frequency sound. Also, of the papers I researched, only a few address low frequencies. Still, even those papers take into account only 1/3-octave bands as low as band number 18 (centered at 63 Hz) while the focus of study tends to be on much higher bands.

The proceeding research describes the main testing apparatus, a concrete waveguide herein called “the chamber,” in Chapter 2. MATLAB code written for using the chamber serves as an operator’s manual for future researchers. All of the code is included and well documented in Chapter 7.

This thesis takes an electrical approach; a pressure-voltage analogy is used to compare the acoustic setting with an electrical one. The waveguide is analogous to an electromagnetic cavity resonator (or electrical transmission line in one dimension), and analogous parameters are laid out in a table in order to make solutions of the wave equation in either analogy straightforward.

The quality factor  $Q$  of the chamber is studied extensively in Chapter 3. Varying amounts of vegetation are placed inside the chamber to affect the  $Q$ . One goal of such tests is to find a relationship between  $Q$  and the absorption coefficient  $\alpha$  of vegetative material. This relationship is still under investigation and could be a subject of future research.

Results for experiments with bales of straw and a full White Pine tree are included in Chapter 4. Gravel, at one time outside of the scope of this paper, is tested in order to compare foliage results with a nonvegetative absorber.

In Chapter 5, the electrical analogy is used to find the impedance and reflection characteristics of anything placed in the chamber. Once again, straw bales and tree branches are used as test foliage. MATLAB code is supplied in an appendix and can be used by future researchers.

Chapter 6 shows results of how a standing wave inside our testing chamber is affected by adding increasing amounts of vegetation to dampen the system.

The chamber handbook is presented in Chapter 7. MATLAB code along with extensive commenting is included so that future researchers can copy and comprehend the routines used herein.

Finally, conclusions are given and a discussion of the limitations of the results obtained is presented. This work does make use of other researchers' testing methods, however it does not attempt recreate their work.

## 2 TESTING FACILITY AND OPERATION

The testing facility is the main component that makes this research original and unique. The concrete chamber described in Section 2.1 could not be recreated without a sizeable budget, an abundance of space, and long-term planning. MATLAB code was written to work with and process the signals taken on the recording device. This recorder is described in detail in Section 2.2. Section 2.3 describes the approach taken in reconciling transmission theory and operating the facilities.

### 2.1 Impedance Chamber

The main testing apparatus for this research is a reinforced concrete waveguide<sup>3</sup> inside the laboratory at the U.S. Army Corps of Engineers Construction Engineering Research Laboratory (CERL). The outside of the chamber is shown in Figure 2.1.



Figure 2.1 Reinforced concrete impedance chamber with steel door.

The chamber's design is such that its 2 m × 2 m cross-sectional dimensions are much smaller than its length of 8.8 m. Thus, the first four modes of oscillation of an acoustic field inside the chamber are the modes (1,0,0), (2,0,0), (3,0,0), and (4,0,0) corresponding to roughly 20 , 40 , 60 , and 80 Hz, respectively. The current research deals with one-dimensional uniform time-harmonic plane waves.

Minimal displacement occurs in the reinforced concrete walls during internal acoustic reflection so coefficients of internal reflection  $\bar{\Gamma}$  (the overbar denotes that  $\bar{\Gamma}$  is complex) approach an ideal value of 1.0 with no phase shift. A door closes at the front of the chamber, allowing items to be placed inside for experiments. A loudspeaker shown in Figure 2.2 is sealed in a small cutout in the center of the wall opposing the door, herein called the back of the chamber. This loudspeaker is the noise source for normal-incidence plane waves and can input a pulse, white noise, pink noise, a pure tone, or any other signal into the chamber.



Figure 2.2 Loudspeaker installed in the center of the back of the chamber.

Microphone ports shown in Figure 2.3 are cut out of the chamber wall at different heights and locations along its length. This research makes use of the ports that are 1 m above the chamber floor (halfway up the 2 m wall). When a steady state pure tone is input to the chamber, these microphones record the pressure fluctuations of a standing pressure wave.



Figure 2.3 The microphone port on the left has a cable running out of it, whereas the other microphone ports are plugged.

## 2.2 Yokogawa ScopeCorder

Although almost any data recorder could be used in these experiments, the Yokogawa ScopeCorder DL750, which is available at the testing facility, is important for four main reasons. First, the DL750 can record at sampling rates up to  $2 \times 10^6$  samples per second. Although such high sampling rates and frequencies are not necessarily needed for the low-frequency scope of this paper, they *are* beneficial for testing code over a large range of conditions. Using such a versatile machine allows future researchers to make use of the code contained in this paper even if their research is targeted at higher frequency bands.

Second, 16 channels are available at one time, making time sync between all microphones in the chamber simple and reliable. Third, the ScopeCorder is flexible in its methods of downloading data digitally for use and processing by computer programs. Specifically, the ScopeCorder has an internal hard drive, an IBM microdrive that holds a 1 GB PC card, and an Ethernet port for input and output of data via a category-5 cable. Fourth, this data-recording device has a large full-color LCD screen on which to view the channels' inputs before, during, and after the data are taken. This is incredibly useful to pinpoint a bad microphone, preamp, power supply, or channel setting thus saving time when debugging faulty systems.

### 2.3 Theoretical Operational Rationale

Taking an electrical approach to an acoustical problem simply means using electromagnetic theory and parameters in the following wave equation,<sup>4, 5, 6, 7</sup> shown here for acoustic plane waves propagating in the  $z$ -direction where  $p$  is pressure,  $c$  is the speed of the wave, and  $t$  is time:

$$\frac{\partial^2 p}{\partial z^2} = \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} \quad (2.1)$$

Table 2.1 shows the parameters of the “pressure-voltage” analogy and their units used throughout the rest of this paper. There are other valid analogies such as the “pressure-current” analogy that will not be covered here.

In order to understand plane wave propagation in the empty chamber, MATLAB code based on transmission line theory was written and is included in Appendix A. The MATLAB function `RUNCHAMBER.m` allows the user to input a frequency of pure tone at which they would like to excite the chamber, initially at rest. It makes use of concepts such as impedance, reflection, Ohm's law, and superposition all while outputting three



graphs: a positive-going digital pressure wave, a negative-going digital pressure wave, and a superimposed total digital pressure wave.

**Table 2.1** The acoustic/electric/mechanical analogy

<i>Acoustic</i>		<i>Electrical</i>		<i>Mechanical</i>	
<i>Parameter</i>	<i>Units</i>	<i>Parameter</i>	<i>Units</i>	<i>Parameter</i>	<i>Units</i>
Compliance	$\frac{m^3}{Pa} = \frac{m^5}{N}$	Capacitance	$\frac{q^2}{N \cdot m}$	Compliance (stiffness <sup>-1</sup> )	$\frac{s^2}{kg}$
Pressure	$Pa = \frac{N}{m^2}$	Voltage	$V = \frac{N \cdot m}{q}$	Force	$N = \frac{kg \cdot m}{s^2}$
Volume	$m^3$	Current	$A = \frac{q}{s}$	Particle Velocity	$\frac{m}{s}$
Velocity	$\frac{m}{s}$	Charge	$q$	Distance	$m$
Volume Displacement	$m^3$	Inductance	$\frac{V \cdot s}{A}$	Mass	$kg$
Inertance <sup>5</sup> or Acoustic Mass <sup>4</sup>	$\frac{N \cdot s^2}{m^5} = \frac{kg}{m^4}$	Impedance	$\Omega$	Mechanical Impedance	$\frac{N}{m/s}$
Impedance	$\frac{Pa \cdot s}{m^3}$	Resistance	$\Omega$	Viscous Friction	$\frac{N}{m/s}$
Resistance	$\frac{Pa \cdot s}{m^3}$				

Figure 2.4 shows a single frame of the moving picture output of **RUNCHAMBER** after the loudspeaker has been turned on at a constant pure tone of 21.5 Hz for 0.5 s (these numbers were chosen arbitrarily). The top graph is the positive-going wave  $P_p$  moving to the right, the middle graph is the negative-going wave  $P_n$  moving to the left, and the bottom graph is the total pressure  $P_{tot}$  in the chamber. A distinct jump in pressure in  $P_n$  can be seen at a distance of 2 m down the chamber axis due to the onset of a reflected wave with the source-produced  $P_p$ . This jump travels back and forth with the traveling waves.

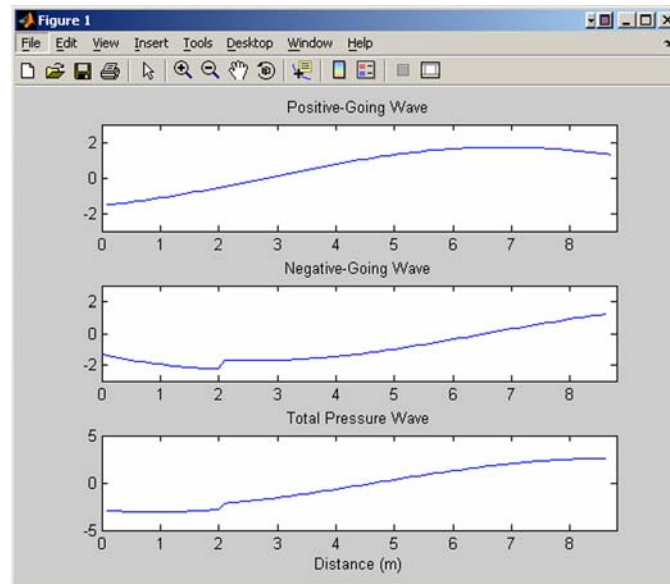


Figure 2.4 A single frame of the output of **RUNCHAMBER**. The actual output is a slower-than-real-time movie where the viewer can track the movement and reflection of pressure in the chamber.

### 3 $Q$ (QUALITY FACTOR)

#### 3.1 Methods of Computing $Q$

In this section, we investigate several methods of computing the  $Q$  of the chamber. Historically,  $Q$  designated the ratio of reactance to resistance of a series circuit element.<sup>8</sup> It applies to mechanical, electrical, and acoustic systems. The letter “Q” was arbitrarily chosen and eventually took on the name “quality factor.”

##### 3.1.1 Method 1: the 3 dB-down method

The definition of  $Q$  that describes a system’s frequency selectivity is

$$Q = \frac{f_r}{\text{Bandwidth}} = \frac{f_r}{f_1 - f_2} \quad (3.1)$$

where  $f_r$  is the resonant frequency of the system and  $f_1$  and  $f_2$  are the  $-3$  dB points (determined by half power) on each side of  $f_r$ . Maximum power delivered by a source to a system occurs at  $f_r$ .

For example, the magnitude of impedance  $|\overline{Z}|$  of the resonant circuit shown in Figure 3.1 is shown in Figure 3.2 as a function of frequency. At low frequencies, voltage across the resistor goes to zero because an inductor is a short at DC.

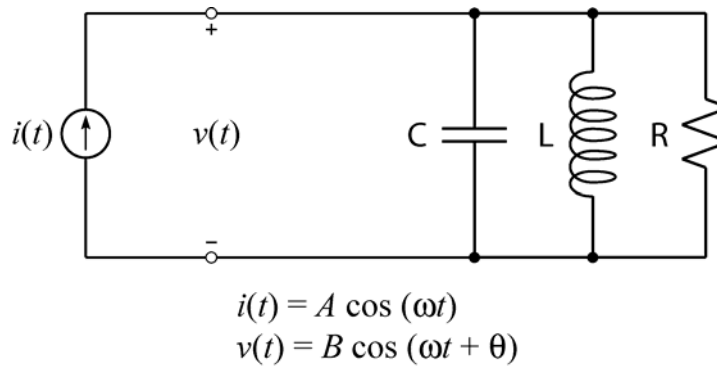


Figure 3.1 A parallel resonant circuit.<sup>9</sup>

At high frequencies, voltage also goes to zero because a capacitor is a short as  $f \rightarrow \infty$ . Maximum power is transferred (dissipated) to the resistor when the inductor and capacitor resonate each other at  $f_r$ . Bandwidth, determined by the frequencies where half peak power is delivered to the resistor, is determined by points where  $|\bar{Z}|$  is  $1/\sqrt{2}$  times the peak value of  $|\bar{Z}|$  on the frequency versus impedance curve shown in Figure 3.2.

Figure 3.2 shows an example of a graphical representation of finding the  $Q$  from the bandwidth of a system.

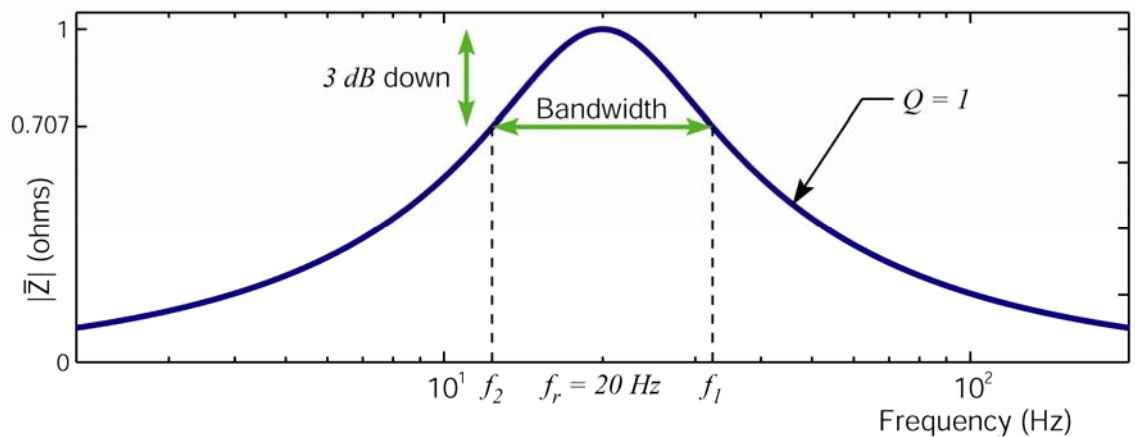


Figure 3.2 The bandwidth of a parallel RLC circuit whose resonant frequency is 20 Hz and where  $Q = 1$ .

Section 3.2 explains in detail how the  $Q$  of an acoustic system can be found using pressure and volume velocity (to obtain impedance as a function of frequency) just as easily as the  $Q$  of an electrical circuit can be found using voltage and current (to obtain impedance as a function of frequency).

### 3.1.2 Method 2: the energy method

Another way  $Q$  can be defined for drivers, driver/box combinations, electromagnetic resonators, acoustic cavities, and other analogous systems is

$$Q = 2\pi \times \frac{\text{(energy stored at resonance)}}{\text{(energy dissipated at resonance)}}, \quad (3.2)^{10, 11}$$

$$Q = \omega \frac{\text{(overall reactive energy in the enclosure)}}{\text{(dissipated power)}}, \quad (3.3)$$

$$Q = 2\pi f \frac{\text{time average of the energy stored in the system}}{\text{rate of energy dissipation}}, \quad (3.4)^{11}$$

$$Q = 2\pi \frac{\text{(oscillating energy)}}{\text{(energy loss per cycle of oscillation)}}, \quad (3.5)$$

$$Q = \omega \frac{\text{peak value of energy stored in 1 cycle}}{\text{dissipated power}}. \quad (3.6)^{12}$$

These definitions are mathematically equal and directly related to the other two methods presented here.

### 3.1.3 Method 3: the logarithmic decrement method

A third way  $Q$  can be defined is via the logarithmic decrement method. The decrement  $\delta$  is defined as the ratio of the amplitude of any damped oscillating peak to the amplitude of the next one.<sup>13</sup> The logarithmic decrement  $\Lambda$  then is defined as

$$\Lambda = \ln(\delta) \quad (3.7)$$

and

$$Q = \frac{\pi}{\Lambda}. \quad (3.8)$$

When the input of a steady state system (such as the one shown in Figure 3.1) is shut off, its response decays logarithmically corresponding to  $e^{-\gamma t}$ , where  $\gamma$  is an attenuation constant and  $t$  is time. An example of this is shown in Figure 3.3.

The relationship between  $Q$  and  $\gamma$  can be found from the following procedure. Let  $A_0$  and  $A_1$  be the amplitudes of successive larger and smaller oscillations, respectively. The decrement is defined as

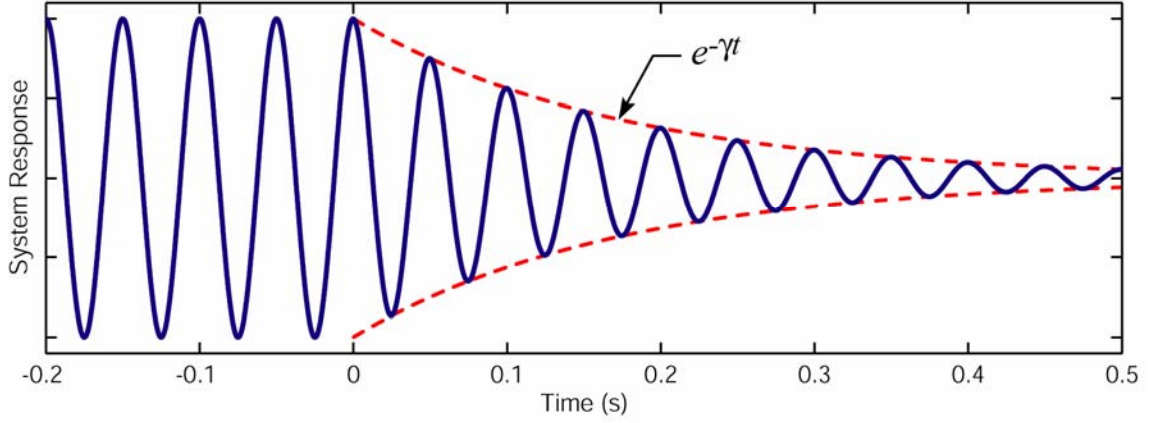


Figure 3.3 This theoretical device has a steady response to a 20 Hz pure tone signal until the input is completely cut off at time  $t=0$  s. The device's energy decays logarithmically.

$$\delta = \frac{A_0}{A_1}. \quad (3.9)$$

Let the period of the system be  $T$ ,

$$T = \frac{1}{f} \quad (3.10)$$

where  $f$  is the frequency of the oscillation. The logarithmically decaying envelope defines the amplitudes of oscillations as follows<sup>5</sup>

$$A = A_0 e^{-\gamma t} \quad (3.11)$$

where  $A$  is the amplitude of the decaying sinusoid at some time  $t$ . For two successive oscillations, occurring one period  $T$  apart, Equation (3.11) can be rewritten as

$$e^{\gamma T} = \frac{A_0}{A_1}.$$

Substituting Equation (3.9),

$$\begin{aligned} e^{\gamma T} &= \delta, \\ e^{\gamma/f} &= \delta. \end{aligned} \quad (3.12)$$

Plugging Equation (3.12) into Equation (3.7), we get

$$\Lambda = \ln \left( e^{\gamma/f} \right),$$

$$\Lambda = \frac{\gamma}{f}. \quad (3.13)$$

Finally, plugging Equation (3.13) into Equation (3.8), we get

$$Q = \frac{\pi f}{\gamma}. \quad (3.14)$$

Mathematically, one implication of this result is that the  $Q$  equals the number of cycles undergone until the amplitude reaches  $e^{-\pi}$  ( $\sim 0.04$ ) of the initial value.

Another equivalent perspective is to measure the amplitude of two consecutive cycles of the decay. The second cycle's amplitude will have a value that is  $e^{-\pi/Q}$  times the amplitude of the previous cycle. If the  $Q$  of the chamber is high, however, this method is difficult with empirical data because the ratio of amplitudes of successive cycles of the decay will be extremely close to 1.

In such instances, it is best to count multiple oscillations of the system. If the decay of amplitude in one cycle of oscillation were too small to measure, one could measure the decay of some number  $N$  cycles of decay. In this case, the  $N^{\text{th}}$  cycle's amplitude will have a value that is  $e^{-N\pi/Q}$  times the amplitude of the original cycle.

### 3.2 Choosing a Method

The first attempt to measure  $Q$  consisted of sending a sweepable pure tone into the empty chamber. A microphone was placed inside the chamber, and its output was displayed on the screen of an oscilloscope. This test did not work well because a high  $Q$  implies a very narrow bandwidth (see Figure 3.2). The analog signal generator was not precise enough to dial in changes of frequency  $\Delta f$  small enough to find such a

bandwidth. The values of the three frequencies needed for Equation (3.1), namely,  $f_r$ ,  $f_1$ , and  $f_2$ , were determined by counting divisions on the screen of the oscilloscope. Although this constitutes a crude method,  $Q$  for mode (1,0,0) was found to be 73 (a value very close to later, more precise tests).

After this initial testing, data was no longer analyzed in real time. Rather, it was recorded and processed in MATLAB using signal processing routines to precisely analyze the signals.

The second experiment to measure  $Q$  attempted to make good use of the aforementioned logarithmic decrement method. Once again, a signal generator with a pure tone at a user-defined frequency was connected to the chamber's loudspeaker. A microphone was placed inside the chamber and this time its output was recorded on a Sony TCD-D8 DAT recorder. The pure tone was played long enough for a steady state standing pressure wave to be set up inside the chamber. The input to the chamber was then abruptly cut off, and the decay of the standing pressure wave was recorded and analyzed.  $Q$  for mode (1,0,0) was 85.

The third type of test came about in order to measure more than one frequency at a time. For this third group of tests, an impulse (or spark) was put into the chamber. An impulse has equal energy across all frequencies. The decay of the pressure wave due to the impulse was too difficult and erratic to analyze with logarithmically decaying envelopes.

The fourth and final way to analyze the chamber was to use white noise as the input source, once again at steady state. White noise is made up of impulses with random amplitude in succession, so it too has equal energy across all frequencies. Since we are only interested in the first four resonant modes of the chamber



(frequencies below 100 Hz), the white noise was lowpass filtered with a cutoff of 110 Hz. Figure 3.4 shows the power spectrum of the filtered white noise.

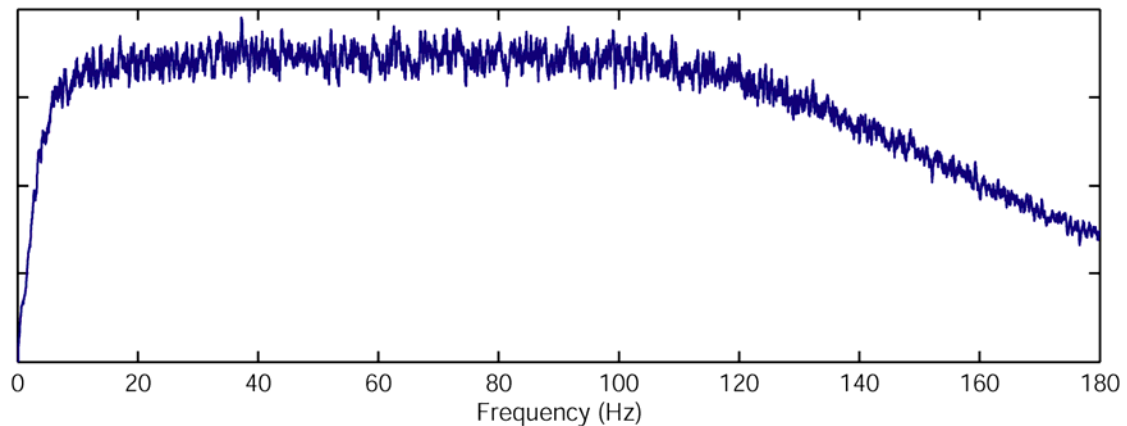


Figure 3.4 The power spectrum of the input white noise is flat in the frequency range of interest, i.e., 15-100 Hz.

Microphones along the chamber wall sent pressure signals due to the white noise inside the chamber to the aforementioned ScopeCorder. Data were sampled at 500 Hz, well above Nyquist frequency for signals in the range of 15-100 Hz. The discrete Fourier transforms (DFTs) of the signals were analyzed. Figure 3.5 shows the result from a single microphone after smoothing algorithms detailed in Appendix B.

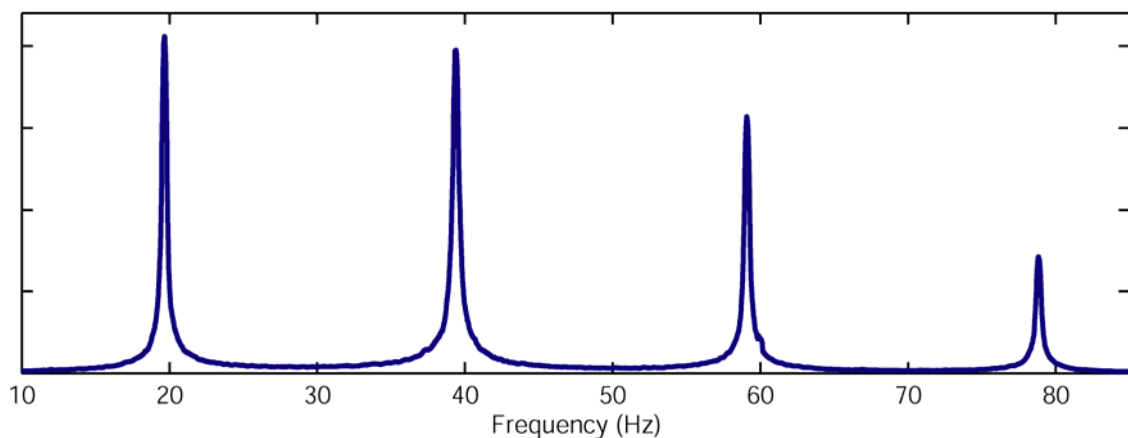


Figure 3.5 The magnitude of the DFT of a microphone in the chamber when white noise is the chamber's input.

The 3-dB down method was used on the peaks in Figure 3.5 to obtain the  $Q$  at each resonance. As long as there are enough points in the DFT shown in Figure 3.5,

the bandwidth of system resonant peaks can easily be found. With enough points so that the frequency resolution  $\Delta f = 1/80$  Hz, this proved to be a reliable and useful method of analysis and is the method used throughout the rest of this paper.

### 3.3 $Q$ and $\alpha$ , Absorption Coefficient

The absorption coefficient  $\alpha$ , defined<sup>14</sup> as the ratio of the energy absorbed by a surface to the energy incident upon a surface, of an absorbing material placed against the door so as to terminate the chamber is a dimensionless constant related to  $Q$ . Beranek<sup>14</sup> equates  $\alpha$  to a measure of the absorbing power of unit area.

In the past, measurement of  $\alpha$  has been a detailed process often performed by laboratories created specifically for such experimental determination. The method most often used consisted of measuring reverberation times in a reverberation chamber. The research in this thesis improves upon their limitations because it deals with steady state signals, thus minimizing error. The particular relationship between  $\alpha$  and  $Q$  is currently under investigation and may be a topic of future research.

### 3.4 Laboratory Techniques

Vegetation placed inside will lower the  $Q$  from that of an empty chamber, however the change is dependent upon the location of the vegetation. This problem can best be studied via the electric analogy.

In transmission line theory dealing with lossless lines varying sinusoidally at steady state, the location of maximum voltage occurs at a location of minimum current (and vice versa).<sup>11</sup> A transmission line terminated by an open circuit at one end and a short circuit at the other end is shown in Figure 3.6 to illustrate one mode of standing

waves and depict boundary conditions that go along with such line terminations. Using Ohm's law ( $Z = V/I$ ), an added impedance (no matter how large) placed at a location where current is zero does *not* affect the standing wave pattern. One could open circuit the line at current nulls and short the line at voltage nulls without changing these natural oscillations.<sup>11</sup>

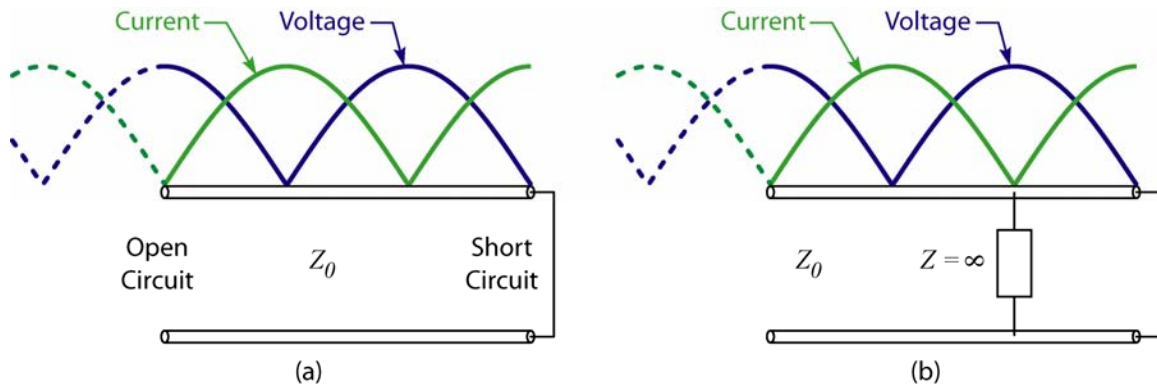


Figure 3.6 Standing current and pressure waves shown on a hypothetical transmission line with characteristic impedance  $Z_0$  (a). The placement of even an infinitely large impedance at a current null does not affect the standing waves (b).

In the acoustic analogy, maximum volume velocity in a transmission medium occurs where the pressure is minimum. An impedance of any magnitude (but infinitesimal thickness) added where the volume velocity is zero (maximum pressure) will *not* affect the acoustic standing waves. Therefore, to study the effect of added impedance on the standing wave patterns, it is necessary to add such impedances (i.e., vegetation mass) where the pressure wave has a null. Due to pressure doubling, the standing pressure waves of the empty chamber at any frequency will have a maximum at the chamber door. The shapes of the first four fundamental modes of the empty chamber are shown in Figure 3.7.

To best study the effect of vegetation on the  $Q$  for mode (1,0,0), the brush would be placed at location  $l/2$ , where  $l$  is the length of the chamber. In the same way, brush

would be placed at  $l/4$ ,  $l/2$ , and  $l/8$  (or any other nulls in the pressure waves shown in Figure 3.7), for modes (2,0,0), (3,0,0), and (4,0,0) respectively.

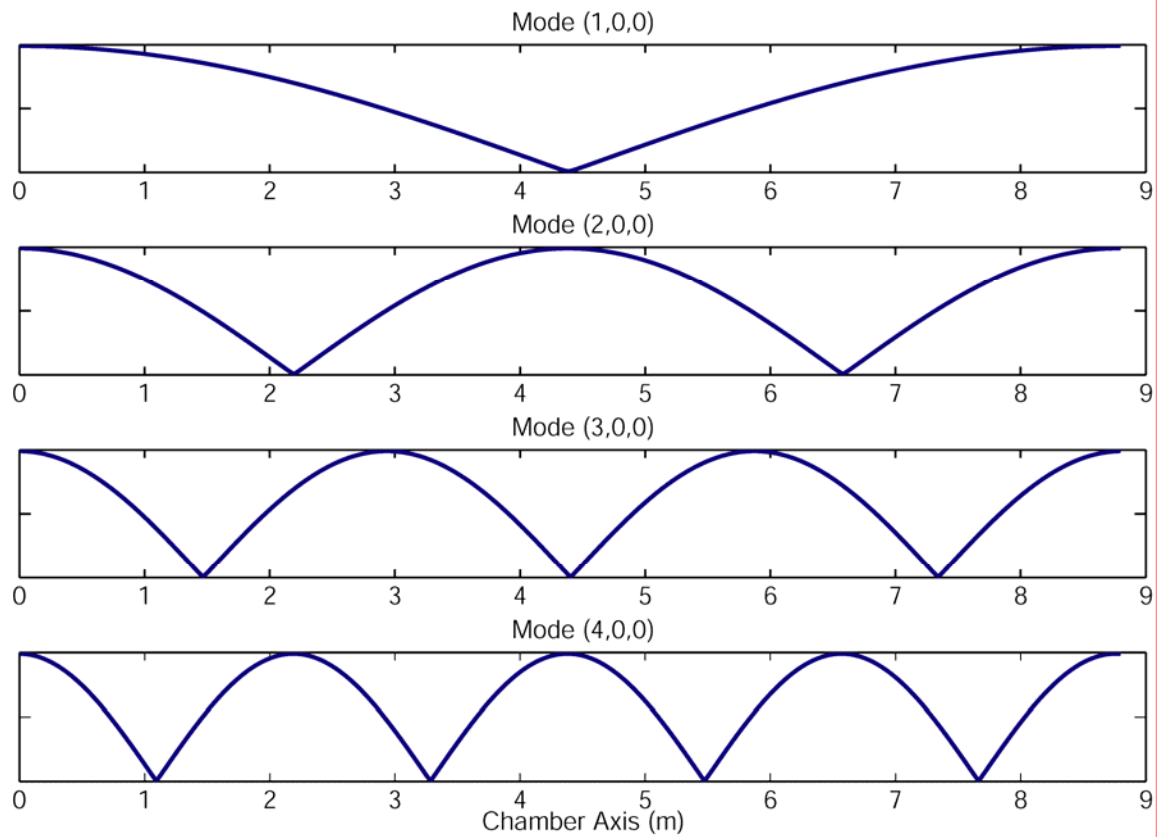


Figure 3.7 Standing pressure waves in an empty chamber for the first four modes. The hard chamber ends are almost perfectly reflecting and can be modeled as an open circuit.

## 4 QUALITY FACTOR RESULTS

### 4.1 Empty Chamber

The empty chamber is the canon to which all other chamber conditions can be compared. Figure 4.1 shows the electrical analogy from which to calculate acoustic parameters in two situations.  $\bar{V}_s$  and  $\bar{Z}_s$  are the AC source voltage and current, respectively.

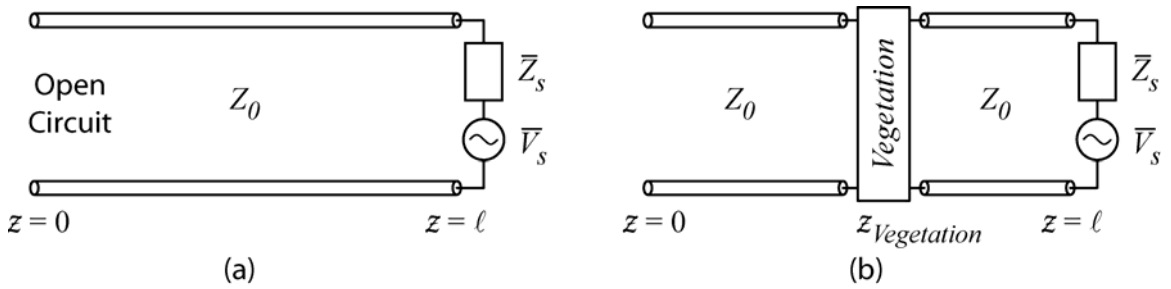


Figure 4.1 The transmission line equivalent of the empty chamber (a) and the chamber with some vegetation placed at some location along its axis (b). The  $z$ -axis runs along the centerline of the chamber. The voltage source represents the loudspeaker (pressure source) in the chamber and the open circuit at  $z=0$  represents the chamber's door.

A 2-port network is used to represent vegetation in Figure 4.1(b). The detailed study of 2-port networks is beyond the scope of this research, but the methods of computing  $Q$  presented in Chapter 3 do not rely on such an analysis. Nonetheless, the model is presented here for future research.

MATLAB code attached in Appendix B makes use of the pressure-voltage analogy to obtain values for  $Q$  and frequencies for the first four fundamental modes of plane wave propagation in any test. Figure 4.2 shows what the microphones measure.

The results for the empty chamber are shown in Table 4.1 on the following page. These numbers are taken from an average of all of the microphones in the chamber.

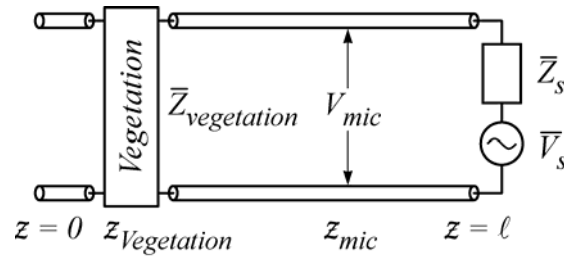


Figure 4.2 The microphones in the chamber measure pressure ( $V_{mic}$  in our analogy) at different locations. Only one microphone's measurement is shown here at the location  $z_{mic}$ . Theoretically, the  $Q$  of the chamber should be the same for every microphone along the chamber axis.

**Table 4.1** Empty chamber  $Q$

<i>Mode</i>	<i>Q</i>	<i>Frequency (Hz)</i>
<i>(1,0,0)</i>	65	19.64
<i>(2,0,0)</i>	114	39.39
<i>(3,0,0)</i>	172	59.13
<i>(4,0,0)</i>	251	78.80

Theoretically, the  $Q$  for an empty chamber with perfectly-reflecting walls should approach  $\infty$ . Any absorption inside the chamber would yield a lower  $Q$ . Hence, the finite results in Table 4.1 show that nonideal conditions were present, although all  $Q > 1$ . Therefore, in the investigation of absorption, it is assumed that the vegetation is the only thing absorbing energy inside the chamber (and the chamber itself does not dissipate energy).

## 4.2 Vegetation in Chamber

Sundry vegetation was placed in the chamber, and quantitative results are included in this section. Each time something is placed inside the chamber, its electrical analogy model can be illustrated as Figure 4.2, where  $\bar{Z}_{vegetation}$  is some unknown impedance due to the vegetation.

#### 4.2.1 Apple tree branches

A 29-kg dry leafless branch of an apple tree was placed in the center of the chamber in an attempt to affect the  $Q$  of the chamber. The branches had no effect (probably due to the low mass and lack of leaves) and the results will not be included here.

#### 4.2.2 Bales of straw

Dense brush more robust than the apple tree branches was needed to affect the  $Q$ , so bales of straw were used. Each bale of straw had a mass of 39.7 kg and measured 0.375 m tall  $\times$  0.5 m wide  $\times$  1 m long. Twenty bales were procured in order to create vegetation walls of different sizes in various locations along the chamber's axis. The resultant  $Q$  for each resonant mode is a function of two variables, namely the location and the amount of added vegetation. For this reason, tests with equally sized straw-bale walls were repeated with the wall in different locations along the chamber axis.

Each bale was situated so that the 0.5 m width was aligned with the  $z$ -axis; thus the area facing the loudspeaker was  $0.375 \text{ m} \times 1 \text{ m} = 0.375 \text{ m}^2$ . The results for straw-bale tests are shown in Figure 4.3.

As predicted by the transmission line analogy, vegetation placed at  $l/2$  affects the  $Q$  for resonant modes (1,0,0) and (3,0,0) more than at any other location (see Figure 4.2(a) and (c)). Likewise, the mode (2,0,0)  $Q$  is affected most substantially by vegetation placed at  $l/4$  (Figure 4.1(b)). Figure 4.2(d) equally supports that mode (4,0,0)  $Q$  is most substantially affected by vegetation placed at  $l/8$ , a standing wave pressure minimum for that mode.

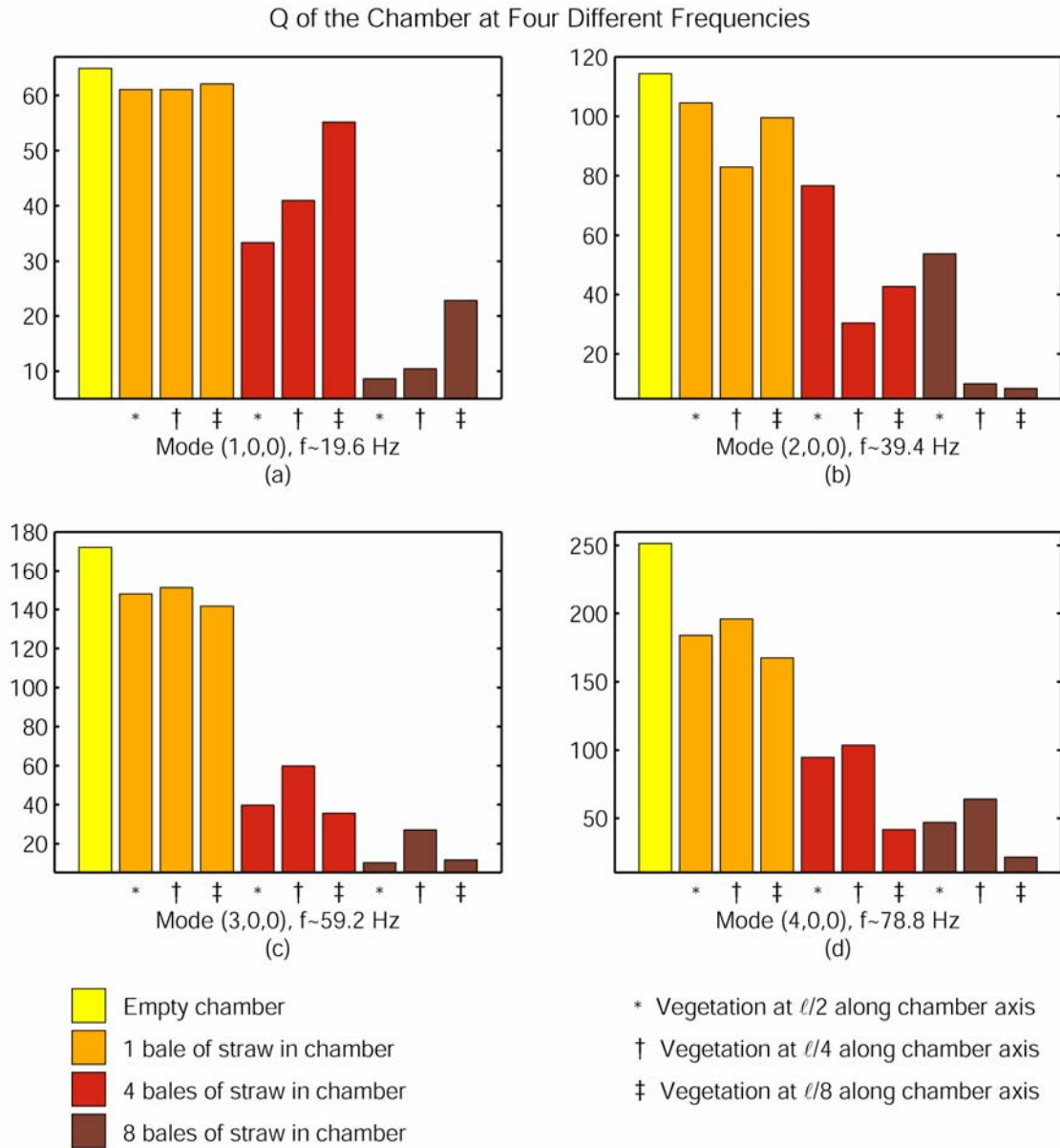


Figure 4.3 For each resonant mode, the  $Q$  of the chamber decreases with increasing vegetation amounts placed inside. Referencing the symbols at the lower right, the placement of the vegetation affects the modes differently as well (see Figure 3.7).

Of interest to some readers and future researchers, Table 4.2 lists the numbers for the graphical representation shown in Figure 4.3. Values where vegetation is placed to theoretically decrease  $Q$  the most are shown in bold font.



**Table 4.2** Chamber  $Q$  values with varying amounts of straw bales

<i>Vegetation Amount</i>	<i>Vegetation Location</i>		
	<i>1/2</i>	<i>1/4</i>	<i>1/8</i>
<b><i>Mode (1,0,0), <math>f \approx 19.6</math> Hz</i></b>			
<i>None</i>	64.90	64.90	64.90
<i>1 bale</i>	<b>61.09</b>	61.09	62.09
<i>4 bales</i>	<b>33.35</b>	41.00	55.16
<i>8 bales</i>	<b>8.60</b>	10.45	22.78
<b><i>Mode (2,0,0), <math>f \approx 39.4</math> Hz</i></b>			
<i>None</i>	114.37	114.37	114.37
<i>1 bale</i>	104.52	<b>82.95</b>	99.54
<i>4 bales</i>	76.69	<b>30.43</b>	42.55
<i>8 bales</i>	53.84	<b>10.03</b>	8.40
<b><i>Mode (3,0,0), <math>f \approx 59.2</math> Hz</i></b>			
<i>None</i>	172.15	172.15	172.15
<i>1 bale</i>	<b>148.03</b>	151.31	141.81
<i>4 bales</i>	<b>39.65</b>	59.83	35.65
<i>8 bales</i>	<b>10.24</b>	27.12	11.64
<b><i>Mode (4,0,0), <math>f \approx 78.8</math> Hz</i></b>			
<i>None</i>	251.49	251.49	251.49
<i>1 bale</i>	183.88	195.99	<b>167.50</b>
<i>4 bales</i>	94.62	103.55	<b>41.72</b>
<i>8 bales</i>	46.99	64.05	<b>21.49</b>

#### 4.2.3 Northern white pine tree

A freshly-cut *Pinus strobes* (White Pine) was brought into the laboratory for testing. The specimen came from the west side of a 15-m thick stand of White Pine at the Vermillion River Observatory in Danville, IL. It was tested in the chamber within two days of being cut.

The 9.2-m tall tree was cut down and driven to CERL on the back of a trailer in two smaller sections totaling less than 9.2 m. The longest section consisted of the top 6 m and 130 lb of the tree. It was bushy with branches full of needles. For this top piece, the trunk measured 5.75 in in diameter at the bottom cut and tapered to a single bunch

of needles at the top. The east-facing side of the tree had significantly fewer branches and needles because it faced the inside of the stand of trees and received less sunlight than the west-facing side.

The second section consisted of the next 1.5 m; the 115 lb section of trunk had a medium amount of branches with needles. The branches on the east side were barren and void of needles. The trunk measured 7 in at the bottom cut.

Two bushy branches were cut from the base of a larger White Pine and piled on the trailer. The branches, shown in Figure 4.4 (a), weighed 89 lb and 70 lb, respectively. They were added simply for additional vegetation mass in the chamber. In addition to the aforementioned 130 lb and 115 lb pieces of the tree, 100 lb of branches that had broken off during tree removal were gathered from the tree and piled on the trailer.



(a)



(b)

Figure 4.4 Two branches (a) cut from another tree were used for extra vegetation mass and the top 130 lb of the tree (b) at the chamber door.

Thus, the total mass from the main tree was (130 lb + 115 lb + 100 lb) 345 lb, with an additional (89 lb + 70 lb) 159 lb of the two branches from another tree. The lowest 1.5 m of the main tree had few branches and needles and was left in the forest.

The main 130 lb section of the tree was loaded into the chamber first, propped on two sawhorses, and is shown in Figure 4.4 (b) on the previous page. The 115 lb section of the trunk was placed in the rear of the chamber (near the speaker).

Unlike the bales of straw in Section 4.2.2, it was impossible to place the tree in discreet locations of standing wave pressure minima. Results for four different amounts of tree are shown in Figure 4.5.

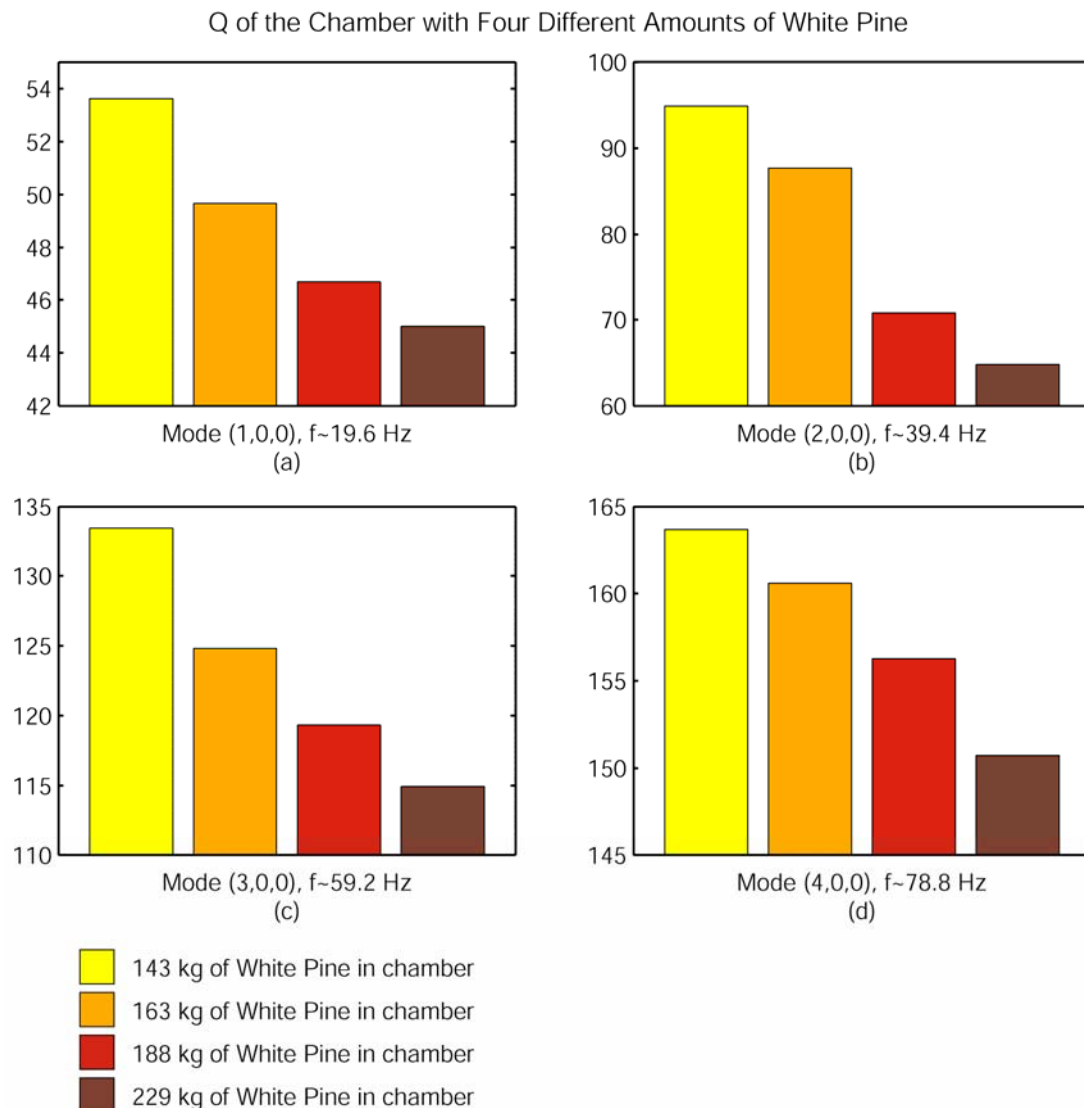


Figure 4.5 Indeed, the  $Q$  is a function of the mass of the vegetation in the chamber.

For each of the first four modes, the higher amounts of tree in the chamber resulted in lower  $Q$ , as expected. It is difficult to determine which parts of the tree were

absorbing energy because the tree did not reside at a finite  $z$ -location. The only way to load the tree into the chamber was along the chamber axis where the tree took up the entire length of the chamber.

Once again, for convenience, the values for  $Q$  are listed in Table 4.3.

**Table 4.3** Chamber  $Q$  values with varying amounts of pine tree

<i>Vegetation Amount</i>	<i>Mode</i>			
	<i>(1,0,0)</i>	<i>(2,0,0)</i>	<i>(3,0,0)</i>	<i>(4,0,0)</i>
<i>None</i>	64.90	114.37	172.15	251.49
<i>143 kg</i>	53.62	94.90	133.46	163.69
<i>163 kg</i>	49.66	87.68	124.82	160.58
<i>188 kg</i>	46.68	70.78	119.33	156.26
<i>229 kg</i>	44.99	64.81	114.93	150.72

### 4.3 Testing Limestone Crushed Rock

Absorption of low-frequency energy is not limited to vegetation. The testing facility and methods detailed here are useful for testing other materials, such as earth or gravel, as well. Three tons of crushed limestone rock were procured and the container called “the box” shown in Figure 4.6 was built.



Figure 4.6 A 2 m × 2 m two-level box.

The lower level of the square box was built using  $2 \times 10$ s; hence it measures 9.25" deep. The upper level of the box was built using  $2 \times 8$ s and measures 7.25" deep. Each side of the box measures 2 m, equal to the width of the chamber.

To continue with the method of incrementally adding mass in the chamber, the first step was to place the lower level of the box inside the chamber as shown in Figure 4.7.



Figure 4.7 The lower level of the box placed at the door side of the chamber.

A test for  $Q$  was also performed on the empty half-box. Ideally, it would have absorbed no energy so that when the box was filled with material to be tested, the decrease in  $Q$  could be attributed solely to that material.

A test for  $Q$  was performed on the empty complete box, shown in Figure 4.8.



Figure 4.8 The full box placed at the door side of the chamber.

Limestone mined in Kankakee, IL, from the Vulcan Materials Company headquartered in Lombard, IL, was delivered to our laboratory. Properties of the CA-7 gravel are shown in Table 4.4.

**Table 4.4** Limestone properties

<i>Property</i>	<i>Value</i>
<i>Soundness</i>	37%
<i>LA Abrasion</i>	28%
<i>Deleterious Chert</i>	0
<i>Soft &amp; Unsound</i>	0.4
<i>Clay</i>	0
<i>Shale</i>	0
<i>Total Deleterious</i>	0.4
<i>Bulk SPG</i>	2.658
<i>SSD SPG</i>	2.702
<i>Absorption</i>	1.6

The rock, shipped from the Urbana transfer yard in 2004 meets all State of Illinois gradation and quality specifications. Gradation information is provided in Table 4.5.

**Table 4.5** Limestone gradation

<i>Sieve</i>	<i>Average</i>	<i><math>\sigma</math>(standard deviation)</i>	<i>Specification</i>
<i>1.5"</i> (37.5)	100	0.0	100-100
<i>1"</i> (25)	100	0.1	90-100
<i>0.75"</i> (19)	89.2	2.7	-
<i>0.625"</i> (16)	72.4	4.4	-
<i>0.5"</i> (12.5)	48.8	4.2	40-56
<i>0.375"</i> (9.5)	23.8	4.2	-
<i>0.25"</i> (6.3)	7.5	2.4	-
<i>#4</i> (4.75)	4.7	1.1	0-10
<i>#16</i> (1.18)	2.2	0.4	-
<i>#200</i> (0.075)	1.76	0.38	0-2.5
<i>Pan</i> (0)	0.00	0.00	-

The lower box, loaded with limestone as shown in Figure 4.9 was tested.





Figure 4.9 The lower half of the box filled with gravel.

Next the full box, loaded with the crushed rock as shown in Figure 4.10 was tested.



Figure 4.10 The full box, fully loaded.

Finally, eight bales of straw were placed on top of the box full of rock in order to further reduce the  $Q$ . The setup is shown in Figure 4.11.



Figure 4.11 Eight bales of straw on top of the full box filled with limestone.

Results for the five box/rock tests are given in Table 4.6. Empty chamber  $Q$  values are given for comparison.

**Table 4.6** Chamber  $Q$  values from rock experiments

<i>Chamber Condition</i>	<i>Mode</i>			
	<i>(1,0,0)</i>	<i>(2,0,0)</i>	<i>(3,0,0)</i>	<i>(4,0,0)</i>
<i>Empty</i>	64.90	114.37	172.15	251.49
<i>Empty lower box</i>	59.18	102.84	169.12	251.58
<i>Empty full box</i>	58.21	99.93	167.42	238.44
<i>Lower box (w/gravel)</i>	57.20	89.29	145.02	171.81
<i>Full box (w/gravel)</i>	49.39	74.39	90.10	127.10
<i>Full box, gravel, and 8 straw bales</i>	32.58	17.83	27.28	19.42

Figure 4.12 shows the graphical representation of the data in Table 4.6 for the sake of consistency.

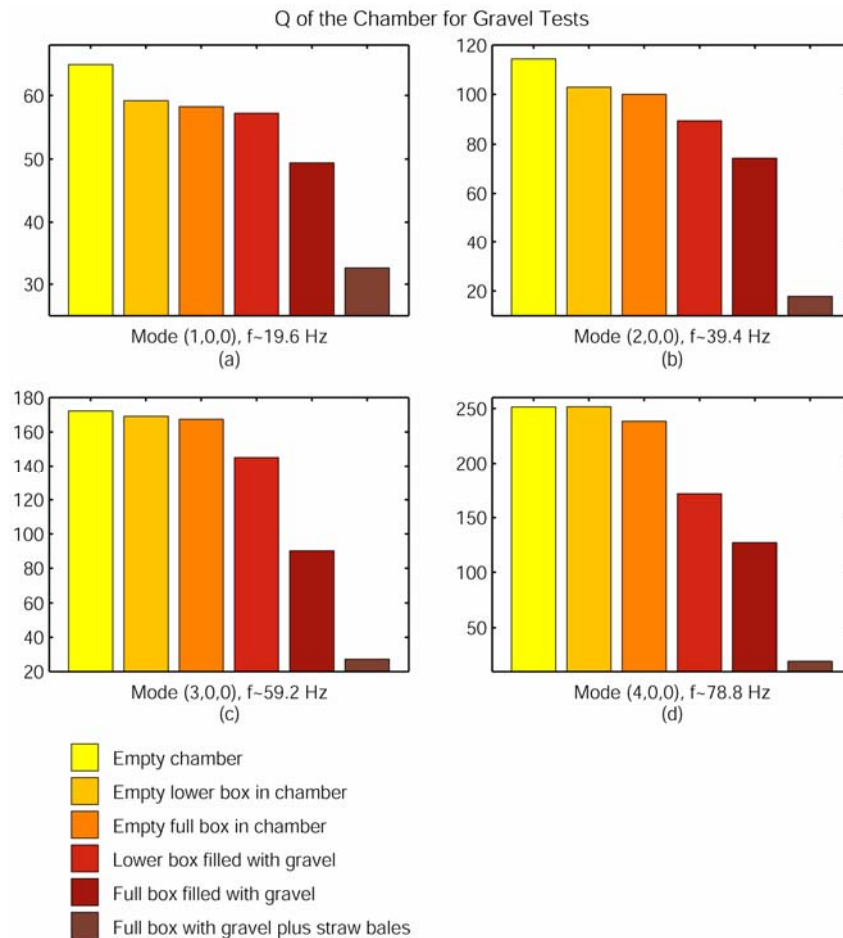


Figure 4.12 The empty chamber compared to the five other tests with varying amounts of rock and the box.



## 5 LOAD IMPEDANCE AND REFLECTION COEFFICIENT

Transmission line theory allows us to find the load impedance  $\bar{Z}_L$  and reflection coefficient  $\bar{\Gamma}_R$  of any substance placed against the chamber door. Unlike Figure 4.2, Figure 5.1 below shows the electrical model with the vegetation terminating the transmission line.

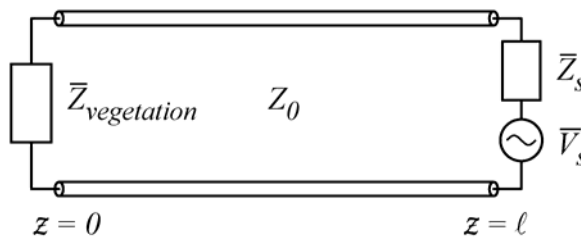


Figure 5.1 The terminating load impedance is most likely complex.

The reflection coefficient may be complex (and introduce phase shift in the reflected wave) when any surface/substance takes the place of the chamber door. The amplitudes of the oscillating pressure waves at each microphone can be used to determine standing wave ratio (SWR),  $\bar{\Gamma}_R$ , and  $\bar{Z}_L$ . The theory and methodology is described below and the explanation of the MATLAB code as a “handbook” to use the chamber is given in Section 7.2.

The speed of sound is found from the temperature  $T_{air}$ , the specific heat ratio ( $k = 1.40$ ), and the gas constant ( $R_{air} = 2.869 \times 10^2$  J/kg/K at the temperature of the laboratory) at that temperature.<sup>6</sup> A controlled temperature of 297 K was maintained throughout the experiments in the laboratory.

$$c_{air} = \sqrt{k R_{air} T_{air}} = 345 \text{ m/s}$$

The pressure stated in height of mercury  $mm_{Hg}$  can be found in the local weather forecast or on the internet. The specific weight of mercury  $\gamma_{Hg}$  is 133 kN/m<sup>3</sup>.

$$P_{atm} = \gamma_{Hg} \cdot mm_{Hg}$$

The static density of air can be found from the following equation:<sup>6</sup>

$$\rho_{0\ air} = \frac{P_{atm}}{R_{air} \cdot T_{air}}$$

The characteristic acoustic impedance inside the chamber can be found and is defined by:<sup>4, 7, 11</sup>

$$Z_0 = \frac{\rho_{0\ air} \cdot c_{air}}{S_{chamber}}$$

where  $S_{chamber} = 4\ m^2$  is the cross-sectional area of the chamber.

In setting up the signal generator, the user may choose a frequency  $f$  at which to excite the chamber with a pure tone. The expression  $2\pi f/c_{air}$  is frequently encountered when using the wave equation. It is convenient to substitute the variable  $\beta$  for said expression:

$$\beta = \frac{2\pi f}{c_{air}}.$$

Ohm's law<sup>5, 11</sup> says

$$Z_0 = \frac{P}{U} = \frac{P_i}{U_i} = -\frac{P_r}{U_r}$$

where  $P$  is pressure and  $U$  is volume velocity. The subscripts  $i$  and  $r$  represent the incident and reflected waves respectively. A simple, general solution<sup>5, 7, 11</sup> to Equation (2.1) is the superposition of a negative-going (incident) and a positive-going (reflected) pressure wave in the coordinate system presented in the previous chapter:

$$P = P_i + P_r$$

Letting the  $z$ -direction be the direction of wave propagation,

$$P(z, t) = P_i \left( t + \frac{z}{c_{air}} \right) + P_r \left( t - \frac{z}{c_{air}} \right). \quad (5.1)$$

$$U = U_i + U_r = \frac{1}{Z_0} (P_i - P_r)$$

The expression corresponding to Equation (5.1) for the phasor chamber pressure<sup>15</sup> is

$$\bar{P}(z) = \bar{P}_i e^{+j\beta z} + \bar{P}_r e^{-j\beta z}$$

where  $\bar{P}_i = |\bar{P}_i| e^{j\theta_p}$ ,  $\bar{P}_r = \bar{\Gamma}_r \bar{P}_i$ , and  $\bar{\Gamma}_r = |\bar{\Gamma}_r| e^{j\theta_r}$ .

$$\begin{aligned} \bar{P}(z) &= \bar{P}_i \left( e^{j\beta z} + \bar{\Gamma}_r e^{-j\beta z} \right) \\ &= |\bar{P}_i| e^{j\theta_p} \left( e^{j\beta z} + |\bar{\Gamma}_r| e^{j\theta_r} e^{-j\beta z} \right) \end{aligned} \quad (5.2)$$

Note that the complex values used in the previous equations are not necessarily equal to their similarly named counterparts. For example,  $\bar{P}_r$  is not the same as  $P_r$ .

The real and imaginary parts of Equation (5.2) are

$$\text{Re}(\bar{P}(z)) = |\bar{P}_i| \cos(\theta_p + \beta z) + |\bar{P}_i| |\bar{\Gamma}_r| \cos(\theta_p + \theta_r - \beta z) \quad (5.3)$$

$$\text{Im}(\bar{P}(z)) = |\bar{P}_i| \sin(\theta_p + \beta z) + |\bar{P}_i| |\bar{\Gamma}_r| \sin(\theta_p + \theta_r - \beta z). \quad (5.4)$$

Because we are interested in steady state conditions, we may choose to let  $t = 0$  when  $\theta_p = 0$ . All phase angles then will be relative to  $\theta_p$ . Setting  $\theta_p = 0$ , Equations (5.3) and (5.4) become

$$\text{Re}(\bar{P}(z)) = |\bar{P}_i| \cos(\beta z) + |\bar{P}_i| |\bar{\Gamma}_r| \cos(\theta_r - \beta z) \quad (5.5)$$

$$\text{Im}(\bar{P}(z)) = |\bar{P}_i| \sin(\beta z) + |\bar{P}_i| |\bar{\Gamma}_r| \sin(\theta_r - \beta z). \quad (5.6)$$

Squaring Equations (5.5) and (5.6):

$$\begin{aligned} \text{Re}^2(\bar{P}(z)) &= |\bar{P}_i|^2 \cos^2(\beta z) + |\bar{P}_i|^2 |\bar{\Gamma}_r|^2 \cos^2(\theta_r - \beta z) \\ &\quad + 2 |\bar{P}_i|^2 |\bar{\Gamma}_r| \cos(\beta z) \cos(\theta_r - \beta z) \end{aligned} \quad (5.7)$$

$$\begin{aligned}\text{Im}^2(\bar{P}(z)) &= |\bar{P}_i|^2 \sin^2(\beta z) + |\bar{P}_i|^2 |\bar{\Gamma}_r|^2 \sin^2(\theta_r - \beta z) \\ &\quad + 2|\bar{P}_i|^2 |\bar{\Gamma}_r| \sin(\beta z) \sin(\theta_r - \beta z)\end{aligned}\quad (5.8)$$

The magnitude of  $\bar{P}(z)$  can be found by plugging Equations (5.7) and (5.8) into the following:

$$\begin{aligned}|\bar{P}(z)| &= \sqrt{\text{Re}^2(\bar{P}(z)) + \text{Im}^2(\bar{P}(z))} \\ \text{Re}^2(\bar{P}(z)) + \text{Im}^2(\bar{P}(z)) &= \\ |\bar{P}_i|^2 + |\bar{P}_i|^2 |\bar{\Gamma}_r|^2 + 2|\bar{P}_i|^2 |\bar{\Gamma}_r| &\left( \cos(\beta z) \cos(\theta_r - \beta z) + \sin(\beta z) \sin(\theta_r - \beta z) \right) \\ = |\bar{P}_i|^2 \left( 1 + |\bar{\Gamma}_r|^2 + 2|\bar{\Gamma}_r| \right) &\left( \cos(\beta z) \cos(\theta_r - \beta z) + \sin(\beta z) \sin(\theta_r - \beta z) \right)\end{aligned}\quad (5.9)$$

An aside:

$$\begin{aligned}\cos(\beta z) \cos(\theta_r - \beta z) + \sin(\beta z) \sin(\theta_r - \beta z) &= \\ \cos(\beta z) (\cos(\theta_r) \cos(\beta z) + \sin(\theta_r) \sin(\beta z)) & \\ + \sin(\beta z) (\sin(\theta_r) \cos(\beta z) - \cos(\theta_r) \sin(\beta z)) & \\ = \cos^2(\beta z) \cos(\theta_r) + 2 \cos(\beta z) \sin(\theta_r) \sin(\beta z) - \sin^2(\beta z) \cos(\theta_r) &\end{aligned}\quad (5.10)$$

and using the identity

$$\cos^2 u - \sin^2 u = \cos 2u$$

where  $u = \beta z$ , we simplify Equation (5.10) to

$$\begin{aligned}\cos(\beta z) \cos(\theta_r - \beta z) + \sin(\beta z) \sin(\theta_r - \beta z) &= \\ \cos(2\beta z) \cos(\theta_r) + 2 \cos(\beta z) \sin(\theta_r) \sin(\beta z) &.\end{aligned}\quad (5.11)$$

Making use of the identity

$$2 \cos(u) \sin(u) = \sin(2u)$$

where  $u = \beta z$ , we further simplify Equation (5.11) to

$$\begin{aligned}\cos(2\beta z) \cos(\theta_r) + 2 \cos(\beta z) \sin(\theta_r) \sin(\beta z) & \\ = \cos(2\beta z) \cos(\theta_r) + \sin(2\beta z) \sin(\theta_r) &.\end{aligned}\quad (5.12)$$

Using the identity

$$\cos u \cos v + \sin u \sin v = \cos(u - v)$$

where  $u = 2\beta z$  and  $v = \theta_r$ , we simplify Equation (5.12) to

$$\cos(2\beta z) \cos(\theta_r) + \sin(2\beta z) \sin(\theta_r) = \cos(\theta_r - 2\beta z). \quad (5.13)$$

Substituting Equation (5.13) into Equation (5.9), we get

$$\begin{aligned} & \text{Re}^2(\bar{P}(z)) + \text{Im}^2(\bar{P}(z)) \\ &= |\bar{P}_i|^2 \left( 1 + |\bar{\Gamma}_r|^2 + 2|\bar{\Gamma}_r| \cos(2\beta z - \theta_r) \right) \\ \therefore |\bar{P}(z)| &= |\bar{P}_i| \sqrt{1 + |\bar{\Gamma}_r|^2 + 2|\bar{\Gamma}_r| \cos(2\beta z - \theta_r)}. \end{aligned} \quad (5.14)$$

Equation (5.14) is the crux of the MATLAB code titled **SW\_FITTER** included in Appendix C and explained in Section 7.2. The unknown variables  $|\bar{P}_i|$ ,  $|\bar{\Gamma}_r|$ , and  $\theta_r$  are determined by minimizing the error between recorded and estimated pressure levels.<sup>16</sup> To simplify the math, they can also be determined by minimizing the error  $E$  between the *square* of recorded levels  $p$  and the *square* of estimated pressure levels defined by

$$\begin{aligned} E &= \sum_{j=1}^N \left( p_j^2 - \left\{ |\bar{P}_i| \sqrt{1 + |\bar{\Gamma}_r|^2 + 2|\bar{\Gamma}_r| \cos(2\beta z_j - \theta_r)} \right\}^2 \right)^2 \\ E &= \sum_{j=1}^N \left( p_j^2 - |\bar{P}_i|^2 \left[ 1 + |\bar{\Gamma}_r|^2 + 2|\bar{\Gamma}_r| \cos(2\beta z_j - \theta_r) \right] \right)^2 \end{aligned} \quad (5.15)$$

where  $p_j$  is the measured sound pressure level at the microphone port with  $z$  coordinates  $z_j$ , and  $N$  is the number of microphones (and thus channels on the recorder) used.

To minimize  $E$ , let the partial derivatives of  $E$  with respect to  $|\bar{P}_i|$ ,  $|\bar{\Gamma}_r|$ , and  $\theta_r$  be zero:

$$\begin{aligned} \frac{\partial E}{\partial |\bar{P}_i|} = & -4 \sum_{j=1}^N \left[ p_j^2 - \left( |\bar{P}_i|^2 \left( 1 + |\bar{\Gamma}_r|^2 + 2|\bar{\Gamma}_r| \cos(2\beta z_j - \theta_r) \right) \right) \right] \\ & \times \left( 1 + |\bar{\Gamma}_r|^2 + 2|\bar{\Gamma}_r| \cos(2\beta z_j - \theta_r) \right) = 0 \end{aligned} \quad (5.16)$$

$$\begin{aligned} \frac{\partial E}{\partial |\bar{\Gamma}_r|} = & -4 \sum_{j=1}^N \left[ p_j^2 - \left( |\bar{P}_i|^2 \left( 1 + |\bar{\Gamma}_r|^2 + 2|\bar{\Gamma}_r| \cos(2\beta z_j - \theta_r) \right) \right) \right] \\ & \times |\bar{P}_i|^2 \left( |\bar{\Gamma}_r| + \cos(2\beta z_j - \theta_r) \right) = 0 \end{aligned} \quad (5.17)$$

$$\begin{aligned} \frac{\partial E}{\partial \theta_r} = & 4 \sum_{j=1}^N \left[ p_j^2 - \left( |\bar{P}_i|^2 \left( 1 + |\bar{\Gamma}_r|^2 + 2|\bar{\Gamma}_r| \cos(2\beta z_j - \theta_r) \right) \right) \right] \\ & \times |\bar{P}_i|^2 |\bar{\Gamma}_r| \sin(2\beta z_j - \theta_r) = 0 \end{aligned} \quad (5.18)$$

Simplifying Equations (5.16)-(5.18),

$$\sum_{j=1}^N \left[ p_j^2 - \left( |\bar{P}_i|^2 \left( 1 + |\bar{\Gamma}_r|^2 + 2|\bar{\Gamma}_r| \cos(2\beta z_j - \theta_r) \right) \right) \right] \left( 1 + |\bar{\Gamma}_r|^2 + 2|\bar{\Gamma}_r| \cos(2\beta z_j - \theta_r) \right) = 0$$

$$\sum_{j=1}^N \left[ p_j^2 - \left( |\bar{P}_i|^2 \left( 1 + |\bar{\Gamma}_r|^2 + 2|\bar{\Gamma}_r| \cos(2\beta z_j - \theta_r) \right) \right) \right] |\bar{P}_i|^2 \left( |\bar{\Gamma}_r| + \cos(2\beta z_j - \theta_r) \right) = 0$$

$$\sum_{j=1}^N \left[ p_j^2 - \left( |\bar{P}_i|^2 \left( 1 + |\bar{\Gamma}_r|^2 + 2|\bar{\Gamma}_r| \cos(2\beta z_j - \theta_r) \right) \right) \right] |\bar{P}_i|^2 |\bar{\Gamma}_r| \sin(2\beta z_j - \theta_r) = 0$$

This set of nonlinear simultaneous equations can be solved using the Newton-Raphson method. Code used for this research included in Appendix C uses a least-mean squares (LMS) algorithm.

Once values for  $|\bar{\Gamma}_r|$ ,  $\theta_r$ , and  $|\bar{P}_i|$  are found,  $|\bar{P}(z)|$  can be calculated with

Equation (5.14). Next, SWR can be calculated with Equation (5.19).<sup>4</sup>

$$SWR = \frac{\text{maximum value of } |\bar{P}(z)|}{\text{minimum value of } |\bar{P}(z)|} \quad (5.19)$$

The variable  $\bar{\Gamma}_r$  can be found by definition as  $\bar{\Gamma}_r = \left| \bar{\Gamma}_r \right| e^{j\theta_r}$ , and  $\bar{Z}_r$  can be found by the following equation:<sup>4, 11</sup>

$$\bar{Z}_r = Z_0 \frac{1 + \bar{\Gamma}_r}{1 - \bar{\Gamma}_r}. \quad (5.20)$$

## 6 STANDING WAVE RESULTS

### 6.1 Vegetation Tests

The chamber was tested in a variety of situations in order to better understand the effect of vegetation. First, the chamber was tested while empty. Next, the chamber was tested with a 3-bale wall of straw (bales stacked on top of each other), a 6-bale wall, and a 10-bale wall termination shown in Figure 6.1 to test the effect of gradually increasing vegetation. Finally, the chamber was tested with 68 lbs. of pine branches and needles hung from the chamber door. This setup is shown in Figure 6.1 below.



Figure 6.1 The chamber terminated with (a) 10-bale straw wall and (b) pine tree branches hung on the inside of the chamber door.

Tests in this section were done at four different pure tone frequencies. The frequencies were chosen at random within the band of interest, namely 20–80 Hz. Both tabular and graphical results are possible, with an example of the latter shown on the following page in Figure 6.2 for the frequency  $f_4 = 70$  Hz. The curves are taken from the output of four different runs of `SW_FITTER.m` mentioned in the previous chapter, explained in detail in Section 7.2, and included in Appendix C.



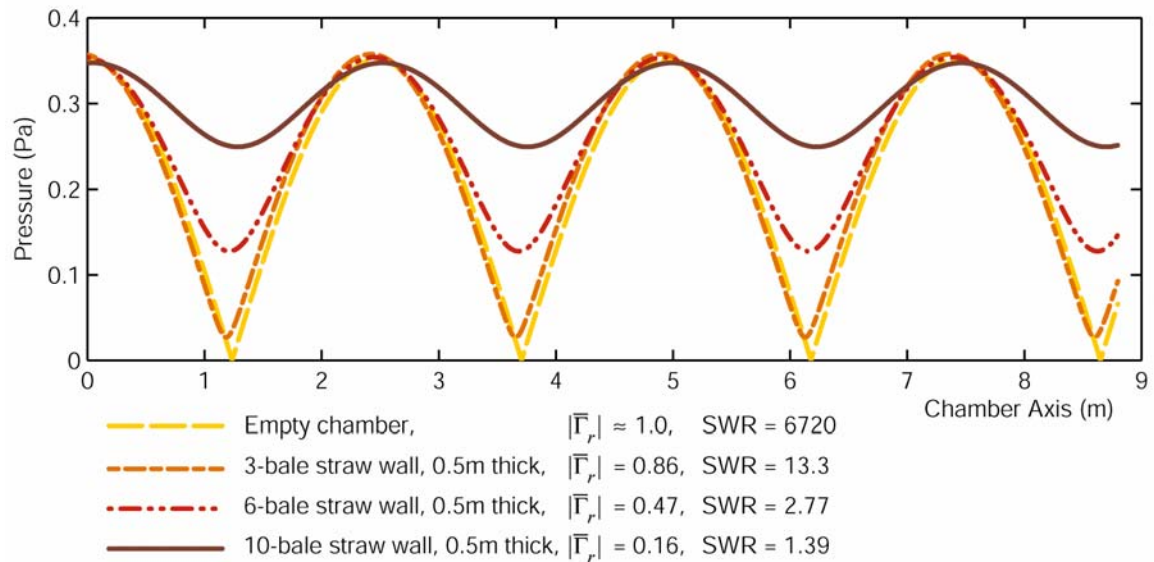


Figure 6.2 Best fit standing wave patterns for a pure tone input at 70 Hz in a chamber terminated with different amounts of vegetation. Resultant values for terminating reflection coefficient and standing wave ratio are also given.

**Table 6.1** Reflection coefficients

Frequency	Terminating Condition	$ \Gamma_r $	SWR
<b><math>f_1 = 25</math> Hz</b>			
	Empty chamber	0.992	249
	3-bale straw wall	0.990	199
	6-bale straw wall	0.906	20.3
	10-bale straw wall	0.914	22.3
<b><math>f_2 = 30</math> Hz</b>			
	Empty chamber	1.00	11 200
	3-bale straw wall	0.934	29.3
	6-bale straw wall	0.932	28.4
	10-bale straw wall	0.848	12.2
<b><math>f_3 = 50</math> Hz</b>			
	Empty chamber	0.97	65.7
	3-bale straw wall	0.888	16.9
	6-bale straw wall	0.740	6.69
	10-bale straw wall	0.524	3.2
<b><math>f_4 = 70</math> Hz</b>			
	Empty chamber	1.00	6 720
	3-bale straw wall	0.860	13.3
	6-bale straw wall	0.470	2.77
	10-bale straw wall	0.164	1.39

Indeed, adding vegetation decreases both the SWR and  $\left|\bar{\Gamma}_r\right|$  as expected. It can also be seen in Figure 6.2 that  $\theta_r$  changes slightly for each case by looking at the change in locations of the peaks of the standing wave.

The case for 68 lbs of pine tree branches (Figure 6.1) is not included in the following results because it did not differ from the results of that of the empty chamber.

From Table 6.1, it appears as though straw absorbs more energy near the higher frequencies (relative to our frequencies of interest, 20–80 Hz).

## 6.2 Gamma Results

Another useful test is to terminate the chamber with some material of interest and plot the  $\left|\bar{\Gamma}_r\right|$  output of `SW_FITTER.m` as a function of frequency. Figure 6.3 shows the results of such a test for three chamber conditions.

The variance in the top line (representing empty chamber conditions) from  $\left|\bar{\Gamma}_r\right|=1.0$  in Figure 6.3 shows the margin of error due to noise in the system, imperfect chamber construction, and the plane wave approximation. The line representing testing of the pine tree-branch wall shown in Figure 6.1 varies with frequency in an unpredictable manner. The wall of branches is so nonhomogeneous that resonances and reflections are unpredictable. The smoothest line gives  $\left|\bar{\Gamma}_r\right|$  for the homogeneous 10-bale straw wall shown in Figure 6.1. It is shown that the bales of straw absorb the most energy in the frequency range of 70-85 Hz.

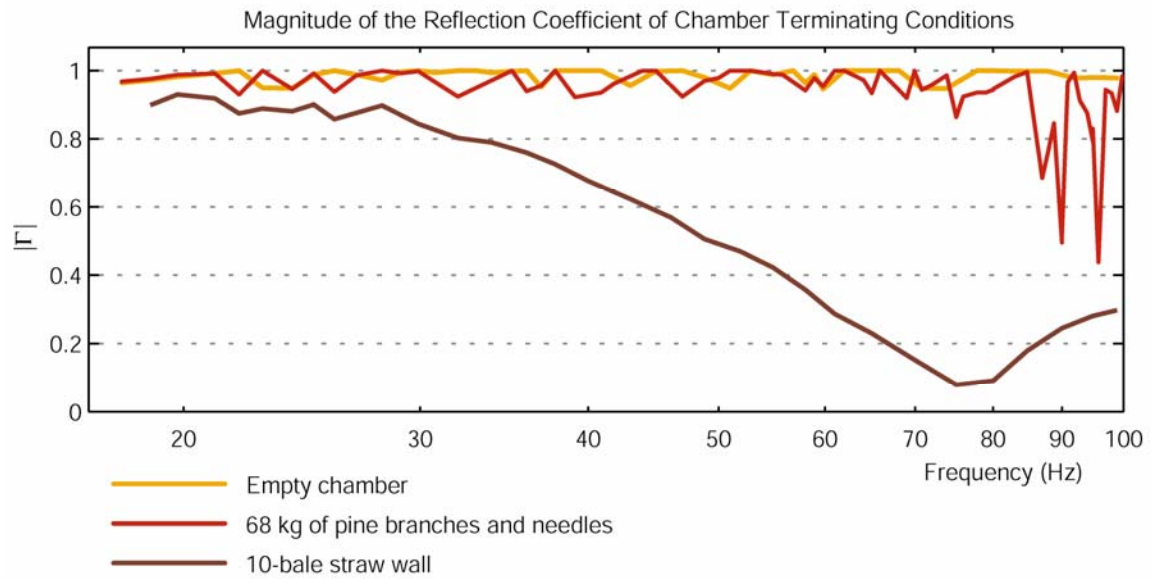


Figure 6.3  $|\Gamma_r|$  as a function of frequency for three chamber terminations: empty, pine tree sections, and bales of straw.

## 7 CHAMBER HANDBOOK

### 7.1 White Noise $Q$

The code explained in this section, titled `white_noise_Q.m` is used to find the  $Q$  of the closed chamber with any vegetation placed inside. It is explained in a piecewise fashion here to act as a handbook for future researchers and is included in its uninterrupted entirety in Appendix B.

The header provides a help menu for the program and explains its limitations. There is only one input parameter, and that must be a character string with the location of a folder from which the user wants to process data.

```
function [frequencies,Qs] = white_noise_Q(folder)
%WHITE_NOISE_Q Calculates Qs of microphones in the chamber at CERL.
%
%   [frequencies,Qs]=WHITE_NOISE_Q(FOLDER) makes use of data
%   taken on the Yokogawa ScopeCorder and prepared by code
%   written by Tim Eggerding at CERL. The folders that hold
%   the shots must be renamed "01", "02", etc. Said folders
%   must be in a main folder identified in the character
%   string FOLDER. Valid for 99 shots or less only.
%
%   There can be between 1 and 16 channels of data taken for
%   any number of shots (or events). WHITE_NOISE_Q goes through
%   FOLDER and averages the DFT of all shots.
%
%   The averaged DFT is then smoothed by bin averaging and the
%   Q is taken using the "3 dB down" bandwidth method.
%
%   The frequencies at which resonant peaks are found are also
%   saved in FREQUENCIES.
%
%   See also YOKOGAWADATAGUI.
%
%   R. Lee, 2004
%   10/01/04
```

The program loads pertinent information and has a variable called `bin_size` that is hard-coded here:

```
load shotTimes           %get the time of all of the shots
num_shots = length(shotTimeArray); %total number of shots to process
H_tot = 0;               %initialize memory
bin_size = 22;           %choose a bin size for bin averaging
                           %this parameter can be changed by the user
                           %at a later point in time based on the
                           %sampling rate and length of time of a
                           %single recorded shot
```

A subroutine included in Appendix B titled `bin_average.m` makes use of `bin_size` to smooth out unwanted jagged edges in a waveform.

If data were taken at a different sampling rate or for a different length of time, the user would most likely change this number to something that gives clear, smooth graphical results. While choosing the right number takes some thought, it is best determined by trial and error and is based on the user's preferences.

Generating noise that is perfectly white is difficult. For short recordings, the spectrum of the output of the random noise generator does not look flat at all. In order to get noise that approaches true random white noise, recordings were excessively long, increasing file size beyond what many computers could process. Taking a larger number of shorter recordings was the solution, and the following `for` loop processes all of those recordings at once:

```
for k = 1:num_shots
    %find out the name of the next folder
    if k < 10 %if it's a single-digit number, like "1", make
               %the folder name a 2-digit number like "01".
        fol_name = sprintf('%s%s','0',num2str(k));
    else
        %else if it's between 10-99, just name the folder
        %the name of that number.
        fol_name = num2str(k);
    end;
    cd(fol_name); %get into that individual folder

    %get sampling rate; it's the same for all channels on the Yokogawa
    cd CH1;load eventHeader;cd ../clear traceName;
    clear date;clear time
    %clear from memory parameters that aren't going to be used

    num_channels = num_folders(cd);%the # of channels we're dealing
                                   %with = the # of folders in the working directory

    %put together data matrix by going into each channel's folder
    %and getting the shot
    data = build_data(num_channels);

    H = abs(fft(data)); %take an DFT of the data matrix
    d_f = samplingRate/size(H,1); %delta F, or the smallest change
                                   %in frequency between points
    freq_axis = 0:d_f:samplingRate-d_f;
                                   %freq_axis is not used in future
                                   %computation, but it is helpful
                                   %to have it here for plotting
                                   %purposes, should you want to
                                   %view the DFT.

    H_av = bin_average(H,bin_size);
```

```
save H H %save variable for later use
```

Here the discrete Fourier transform (DFT) of the shorter recording is stored:

```
save H_av H_av
H_tot = H_tot + H_av; %add up the spectra
cd .. %return to parent directory
end;
```

The DFT, stored in `H_tot` ends up with units that are multiple times too large from all of the smaller shots' DFTs being compounded on top of one another. The following line of code averages out `H_tot` and gives it proper units:

```
H_tot = H_tot/num_shots; %average 'em out
%make a window
l_window = round(2*samplingRate); %length in # of samples
```

The variable `l_window` is a moving set of samples that will parse through the total DFT, `H_tot` to find peaks. Ultimately, these peaks tell us the frequency of the tone being played in the chamber.

```
cutoff_freq = 100; %cutoff, in Hz, above which we
%don't care about peaks
```

A 100 Hz cutoff is instated so that the code does not have to iterate through frequencies in which the user is not interested.

```
num_iterate = round(cutoff_freq/d_f)-l_window+1; %# of times our
%window will have to be run through
maxx = max(H_tot(1:l_window,:)); %a list of maxima from each window
```

As the window parses the DFT, it stores maximum values out of those samples in the window's range.

```
indicator = ones(1,num_channels); %initialize memory
%has range of [1:l_window]
locator = ones(1,num_channels); %locates a point
for k = 2:l_window
    maxx(k,:) = max(H_tot(k:k+l_window-1,:));
    %the maximum value amongst the
    %nearest samples
    for i_mic = 1:num_channels
        if maxx(k,i_mic) == maxx(k-1,i_mic) %if the maximum in this
            %set of samples is the same as the maximum of
            %the last set of samples (one window before)
            indicator(i_mic) = indicator(i_mic) + 1;
            %increment indicator
        else
            indicator(i_mic) = 1; %if there's a new maximum,
            %reset the indicator to 1.
        end;
    end;
end;
```

```

        end;
    end;

    %at this point through (one run through with the window), the only
    %possible local max could be at zero Hz (D.C.)
    for i_mic = 1:num_channels
        if indicator(i_mic) == 1_window           %if so, |H(1,i_column)|

```

If a maximum value in a window stays the same as the window incrementally moves across the DFT's samples, it must be a local peak. If, however, new maxima occur as the window moves, the code is not finding any peaks. When that occurs, the variable `indicator` is reset. The `indicator` is also reset if a local peak has been determined so that another peak could be found:

```

        local_max(i_mic) = H_tot(1,i_mic); %is a local max
        frequencies(i_mic) = 0;           %zero for D.C.
        locator(i_mic) = locator(i_mic) + 1; %increment it

```

Each time a peak is found, `locator` stores the index so that the frequency of that peak can be determined:

```

        indicator(i_mic) = 1;               %reset indicator
    end;
end;

for i_mic = 1:num_channels %for each microphone
    for k = 1_window+1:num_iterate %iterate through the DFT
        maxx(k,i_mic) = max(H_tot(k:k+1_window-1,i_mic));
                                %max value in that window
        if maxx(k,i_mic) == maxx(k-1,i_mic) %if this window's maxx
            %is the same as the last window's, increment the indicator
            indicator(i_mic) = indicator(i_mic) + 1;
        else %else reset the indicator
            indicator(i_mic) = 1;
        end;
        if indicator(i_mic) == 1_window;
            %if so, then |H(1,i_column)| is a local max

```

When a peak is found, it is stored:

```

        local_max(locator(i_mic),i_mic) = H_tot(k,i_mic);
        frequencies(locator(i_mic),i_mic) = freq_axis(k);
        %mark the frequency (Hz)
        locator(i_mic) = locator(i_mic) + 1; %increment it
        indicator(i_mic) = 1; %reset indicator
        indices(locator(i_mic)-1,i_mic) = k; %index in the FFT
    end;
end;
end;

```

When peaks are known, then points determining bandwidth are found:

```

    %bandwidth points are at 1/sqrt(2) times original peak
    BW_point = local_max/sqrt(2);
    value = local_max; %a copy of local_max so we can augment it

```

With the bandwidth points found, the frequencies at those points can be found. In this program they are called `f_low` and `f_high` to represent  $f_2$  and  $f_1$ , respectively, of

Equation (3.1):

```
%get f_low for the "3 dB down" method
for i_mode = 1:size(frequencies,1)%do this for each mode found
    for i_mic = 1:num_channels %do it for each channel
        offset = 0; %an offset (in samples) from f_0
        while value(i_mode,i_mic) > BW_point(i_mode,i_mic)
            value(i_mode,i_mic) = H_tot(indices(i_mode,i_mic)-
                                   (offset+1),i_mic);
            offset = offset + 1;
%increment the offset as long as you're still above the 3 dB down point
        end;
        if indices(i_mode,i_mic)-offset >= 1
            %if the index were ever recorded,
            %do some interpolation to calculate a low frequency to mark
            %the bandwidth point for the "3 dB down" method"
            slope = H_tot(indices(i_mode,i_mic)-offset+1,i_mic) -
                    H_tot(indices(i_mode,i_mic)-offset,i_mic);
            rise = BW_point(i_mode,i_mic)-H_tot(indices(i_mode,i_mic)-
            offset,i_mic);
            f_low(i_mode,i_mic) = (rise/slope+indices(i_mode,i_mic)-
            offset)*d_f;
        end;
    end;
end;

%get f_high for the "3 dB down" method
value = local_max;
for i_mode = 1:size(frequencies,1)%do this for each mode found
    for i_mic = 1:num_channels %do it for each channel
        offset = 0; %an offset (in samples) from f_0
        while value(i_mode,i_mic) > BW_point(i_mode,i_mic)
            value(i_mode,i_mic) =
                H_tot(indices(i_mode,i_mic)+(offset+1),i_mic);
            offset = offset + 1;
%increment the offset as long as you're still above the 3 dB down point
        end;
        if indices(i_mode,i_mic)+offset-1 >= 1
            %if the index were ever recorded,
            %do some interpolation to calculate a high frequency to mark
            %the bandwidth point for the "3 dB down" method"
            slope = H_tot(indices(i_mode,i_mic)+offset,i_mic) -
                    H_tot(indices(i_mode,i_mic)+offset-1,i_mic);
            rise = BW_point(i_mode,i_mic)-
                    H_tot(indices(i_mode,i_mic)+offset-1,i_mic);
            f_high(i_mode,i_mic) =
                (rise/slope+indices(i_mode,i_mic)+offset-1)*d_f;
        end;
    end;
end;
end;
```

Equation (3.1) is used to calculate  $Q$ :

```
Qs = frequencies./(f_high-f_low);%the definition of the 3 dB down Q
Return
```



## 7.2 $\Gamma$ Computation and Standing Wave Fitter

The code explained in this section is used to find information about material placed so as to terminate the chamber. The terms *channel* and *microphone* are often used interchangeably since each microphone's output is recorded on its own separate channel of the recorder. Figure 7.1 page shows an example of bales of straw that were used to test and develop this code.

Acoustic properties of interest are  $\bar{\Gamma}_r$ , SWR, and  $\bar{Z}_L$ . The code makes use of a pure tone put into the chamber with a signal generator and the microphones in the chamber receiving oscillating standing pressure waves.

The frequency of the pure tone is determined, the amplitude of the standing wave at each microphone is calculated, and a standing pressure wave is "best fit" to the data. Figure 7.2 illustrates one example of the output of the code.



Figure 7.1 Ten bales of straw placed to terminate the empty chamber.

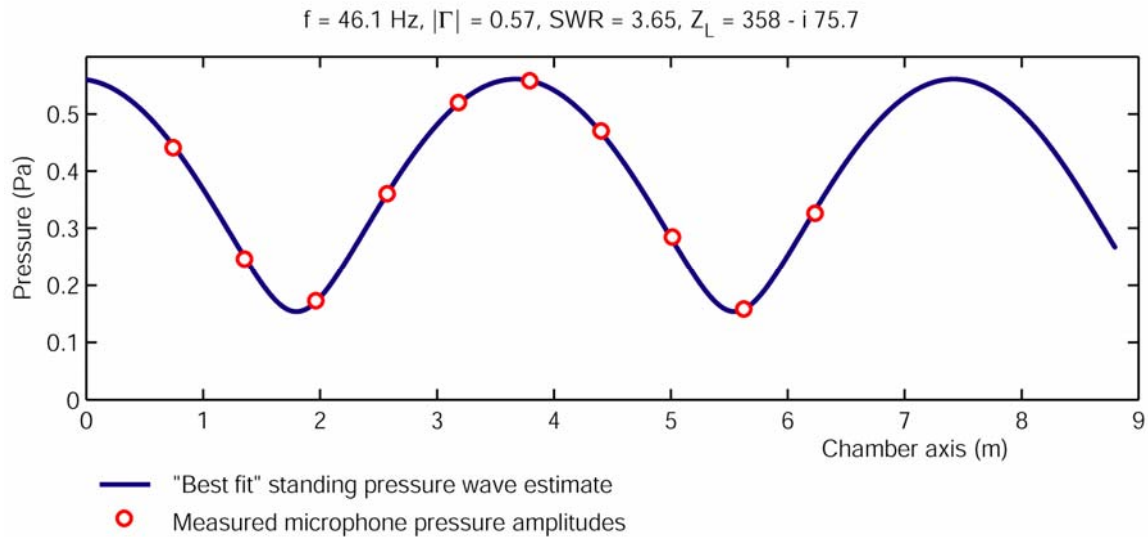


Figure 7.2 The code outputs the best fit standing pressure wave as well as the data given at the top of the graphic: tone frequency, reflection coefficient, standing wave ratio (SWR), and load impedance.

The code `sw_fitter` is explained in a piecewise fashion. It is included in its uninterrupted entirety in Appendix C for neatness. The program begins by defining some constants:

```
%%% DEFINE CONSTANTS
location_rate = 1000;           %samples per meter
l_chamber = 8.798;              %length of chamber in meters
chamber_axis = 0:1/location_rate:l_chamber;
```

This axis is an x-axis with which to plot the output shown in Figure 7.2:

```
T_air = 297;                    %air temp in Kelvin
R_air = 2.869E2;                %Gas constant, (J/kg/K)
k_air = 1.40;                   %specific heat ratio
mm_HG = 762;                    %mm of HG for barometric pressure; 762 mm = 30 in.
gamma_HG = 133;                 %specific weight of mercury (kN/m^3)
```

The constants shown above are true for laboratory conditions at CERL in Champaign, IL. The following equation<sup>6</sup> is for measuring pressure relative to atmospheric pressure (gage pressure):

```
p_atm = gamma_HG*mm_HG;        %atmospheric pressure
```

The following equation<sup>9</sup> is known as the *ideal gas law* and is used to find the density of air:

```
rho_air = p_atm/(R_air*T_air); %kg/m^3
```

The following equation<sup>6</sup> makes use of the ideal gas law to find the speed of sound in air,  $c_{air}$ , at the temperature  $T_{air}$ :

```
c_air = sqrt(k_air*R_air*T_air); %Eq. 1.20 in Munson
```

The characteristic acoustic impedance is defined here:

```
z_0 = rho_air*c_air/4; %Z = rho*c/A where A = 4 m^2
```

Loading the variable `eventHeader` from one of the microphone's folders retrieves the sampling rate of all data taken for this event. The Yokogawa DL750 ScopeCorder can only record its channels with one sampling rate; therefore, retrieving the sampling rate for any single channel suffices for all channels. In the code below, the data taken on channel 1 in the folder "CH1" is used to retrieve the sampling rate for all channels. If, for some reason, channel 1 were not used on the DL750, this code would have to be modified to retrieve the sampling rate from a different channel's folder.

```
%get sampling rate
cd CH1;load eventHeader;cd ..;clear traceName;clear date;clear time
```

The variables `traceName`, `date`, and `time` are not needed for the rest of the program and can be cleared to save memory.

```
loc=[0.744 1.354 1.964 2.574 3.184 3.794 4.404 5.014 5.624 6.234]';
%locations of microphones
```

The variable `loc` lists the  $z$ -coordinate of the first 10 microphones. It should be modified if a different number of microphones are used or some microphones are omitted:

```
%% OPEN THE FILES AND READ 'EM IN
num_channels = num_folders(cd);
%the # of channels we're dealing with =
% # of folders in the working directory
```

The chamber has 13 microphone ports along its length that measure pressure, however Figure 7.2 shows only 10. When all 13 channels were used, `sw_fitter` gave an output where the red circles in Figure 7.2 did not necessarily lie on

the blue “best fit” standing wave. The error was unacceptable and can be explained by higher-order evanescent modes near the loudspeaker.

Three-dimensional modes decay with distance from the speaker as opposed to the  $(n,0,0)$  modes (where  $n$  is the number of half wavelengths shown in Figure 3.7) that set up standing waves. The math presented in Chapter 5 is based on Equation (2.1), which describes a plane wave propagating in only one direction. Since no loudspeaker can create a perfectly plane wave, however, microphones closest to the speaker pick up decaying higher-order modes. Trial and error found that data from these microphones introduced the unacceptable error in `sw_fitter`. Data taken with only the first 10 microphones yielded results that were acceptable, predictable, and agreed with one-dimensional wave theory:

```
%put together data matrix by going into each
%channel's folder and getting the shot
data = build_data(num_channels);
%outputs an array of peak-peak amplitudes and frequencies
```

The subroutine `build_data` is included in Appendix B and puts together a matrix that contains all of the microphones' data in units of pascals. Each channel's data is placed in columns lined up in numerical order.

```
%% FIND THE FREQUENCY OF THE EXCITATION
[amplitude,freq]=freq_finder(data,samplingRate);
```

The subroutine `freq_finder` takes the data matrix and outputs a list of amplitudes (one for each microphone) and a single frequency. The frequency is that of the pure tone being put into the chamber. The amplitudes are the peak values, in pascals, of the sinusoidal oscillating pressure from the standing pressure wave at each microphone.

These amplitudes can be seen as red circles in Figure 7.2:

```
beta = 2*pi*freq/c_air;           %in radians

%% MAKE SOME INITIAL GUESSES
amp = sort(amplitude(:),'descend');
%tabulate all of the various amplitudes
%make the largest mic's amplitude your initial guess
abs_Pi = amp(1);clear amp;
```

The variable `abs_Pi` is a good initial guess for  $|\overline{P}_i|$  and is taken from the largest value in the array `amplitude`. The variable `d_pi` represents  $\Delta|\overline{P}_i|$ , a small increment between guesses of  $|\overline{P}_i|$ .

```
%an incremental amount to differentiate guesses for the magnitude
%of the incident pressure wave
d_Pi = abs_Pi/50;
%initial array of a bunch of guesses for |Pi|
Pi = 0.2*abs_Pi:d_Pi:1.2*abs_Pi-d_Pi;
```

There is also a variable for  $\Delta\theta_r$ , the increment between guesses of  $\theta_r$ , and  $\Delta|\overline{\Gamma}_r|$ , the increment between guesses of  $|\overline{\Gamma}_r|$ .

```
%an incremental amount to differentiate guesses of theta,
%the phase angle of gamma
d_theta = pi/50;
%initial array of a bunch of guesses for theta
theta = d_theta:d_theta:2*pi;
%an incremental amount to differentiate guesses of magnitude of
%gamma, the reflection coefficient
d_gamma = 0.002;
```

The user can replace the very first `d_gamma` in the following line with a larger number by the user to make the following `for` loops run more quickly. Doing so would yield the correct result only if the resultant  $|\overline{\Gamma}_r|$  were between the new number and 1:

```
%initial array of a bunch of guesses for |gamma|
gamma = d_gamma:d_gamma:1;
```

The next three `for` loops create a matrix, `p`, representing  $|\overline{P}(z)|$  given in Equation (5.14). The values in `p` come from combinations of guesses for  $|\overline{\Gamma}_r|$ ,  $\theta_r$ , and  $|\overline{P}_i|$  at each mic location  $z_{mic}$  given in the code by the variable `loc`:

```
%the following 3 for loops generate a 4-dimensional matrix
%consisting of all of the combinations of gamma, theta,
%mic location, and Pi amplitude you can get
for i_gamma = 1:length(gamma)
    for i_Pi = 1:length(Pi)
        for i_theta = 1:length(theta)
            p(i_theta,:,i_gamma,i_Pi) = Pi(i_Pi)*
                sqrt(1+gamma(i_gamma)^2+2*gamma(i_gamma)
                    *cos(2*beta*loc-theta(i_theta)))';
        end;
    end;
end;
```

```
end;
```

A matrix **e** is created to measure the error between **p** and the actual measured pressure amplitudes at each mic location:

```
%create an error matrix
for i_gamma = 1:length(gamma)
    for i_Pi = 1:length(Pi)
        for i_theta = 1:length(theta)
            e(i_theta,:,i_gamma,i_Pi) = amplitude -
                p(i_theta,:,i_gamma,i_Pi);
        end;
    end;
end;
```

The error, **e** is squared to make all values positive:

```
e_2 = e.^2;          %error squared
clear e;
```

A matrix, **error** is created after summing values of **e\_2** over all microphones (values of  $z_{mic}$ ). In other words, each entry in **error** contains the sum of all of the microphone's errors for that particular guess of  $|\bar{\Gamma}_r|$ ,  $\theta_r$ , and  $|\bar{P}_i|$ . It will be one dimension less than **e** or **e\_2** since those matrices have entries for each individual microphone:

```
error = sum(e_2,2); %create an error matrix that's 1 dimension
                  %less than p summing 10 individual errors
clear e_2; clear p
```

The minimum value of **error** occurs at some particular guess of  $|\bar{\Gamma}_r|$ ,  $\theta_r$ , and  $|\bar{P}_i|$ .

Since **error** does not hold the actual value of  $|\bar{\Gamma}_r|$ ,  $\theta_r$ , or  $|\bar{P}_i|$ , the following code finds the index (the location in the matrix **error**) of the minimum value:

```
[throw_away,theta_index] = min(min(min(error,[],4),[],3)); %throw
away the actual value, but keep the index
[throw_away,Pi_index] = min(min(min(error),[],3)); %throw away the
actual value, but keep the index
[throw_away,gamma_index] = min(min(min(error),[],4));
```

Once the index for each of those variables is found, the actual value for the variable can be found by locating the index in the variable array **theta**, **Pi**, or **gamma**, the list of guesses for  $\theta_r$ ,  $|\bar{P}_i|$ , and  $|\bar{\Gamma}_r|$ , respectively:

```
theta = theta(theta_index);      %LMS result for theta
```

```
Pi = Pi(Pi_index);           %LMS result for |Pi| amplitude
gamma = gamma(gamma_index); %LMS result for |gamma|
```

Once the values are found for the variables that will give the best fit curve for  $\left|\bar{P}(z)\right|$ ,  $p$  is

recalculated, this time as a single waveform rather than a matrix of guesses:

```
p = Pi*sqrt(1+gamma^2+2*gamma*cos(2*beta*chamber_axis-theta));
```

The maximum and minimum values of the curve  $p$  yield the SWR in the following line of code:

```
SWR = max(p)/min(p); %standing wave ratio
clear i;
```

The following line of code gives a single complex number to represent  $\bar{\Gamma}_r$  from  $\left|\bar{\Gamma}_r\right|$  and

$\theta_r$ :

```
gamma_i = gamma*exp(i*theta); %define complex reflection coefficient
```

The variable for  $\bar{\Gamma}_r$  can, in turn, be used to find the terminating load impedance,  $\bar{Z}_r$  from Equation (5.20):

```
Z_R = Z_0*(1+gamma_i)/(1-gamma_i); %kg/m^2/s
```

The final step is to be able to create a graphical output similar to Figure 7.2:

```
plot(chamber_axis,p)
hold;plot(loc,amplitude,'ro');
xlabel('Chamber axis (m)');
ylabel('Pressure (Pa)');

title(sprintf('%s%0.3g%s%0.3g%s%0.3g%s%0.3g%s',
    'f = ',freq,' Hz, Gamma = ',gamma,', SWR = ',SWR,
    ', Z_L = ',real(Z_R),' + ',imag(Z_R),'i.'))
```

Each time the code is run, the results are saved as MATLAB `.mat` files as well as a graphical Adobe Illustrator `.ai` file:

```
save p_wave p;
save chamber_axis chamber_axis;
print -dill standing_wave;
return
```

## 8 CONCLUSION

A new method of testing vegetation in a controlled laboratory environment has been developed at the U.S. Army Corps of Engineers CERL. Low frequency absorption and reflection coefficients have been found reliably and consistently. Due to study of the steady state conditions, the methods presented here could constitute a more consistent method than ever before. While there are innumerable species that were not tested, the procedures described in this paper serve as a handbook of sorts for other CERL employees.

One suggested area of improvement would be regarding the third attempt to measure  $Q$  in Section 3.2. Instead of analyzing a decaying signal with all decaying modes together, the signal should be filtered for a single mode and then studied. Although the author believes this method would work, it defeats the benefit of using an impulse as input to the chamber because each resonant  $Q$  would be studied separately.

Another area of improvement would be to incorporate the two-port network presented the electrical model in Figure 4.1 into theoretical calculations. This paper had extremely reliable results when based off of theory, and the acoustic properties of vegetation could be more closely modeled if the parameters of a two-port network were known.

The largest leap of improvement would be to find a definitive relationship between the  $Q$  of the chamber and the absorption coefficient  $\alpha$  of some material. Through many attempts at derivations, discussions with other professors and professionals, and a number of different mathematical representations of the chamber, I believe that such a relationship is not far away.



A suggestion for improvement would be to study the impedance characteristics of the loudspeaker and incorporate them into the electrical model. Once again, theoretical calculations would improve and make the entire setup more reliable and easily understood.

## APPENDIX A

### THEORETICAL STANDING WAVE VIEWER

The following MATLAB code was written to obtain the graphical results shown in Figure 2.4. It is well-commented and uses the most basic transmission equations. The biggest benefit of this program is to be able to watch in slower-than-real-time traveling waves interfere to create a standing wave. One can see the system go from the onset of an input signal all the way to a steady state condition.

```
function [p]=runchamber(f_sound,R_door,R_src,T_air)
%[p] = RUNCHAMBER(f_sound,R_door,R_src,T_air)
%
%   RUNCHAMBER simulates a loudspeaker with a pure tone as the
%   source driving the impedance chamber we have at CERL.
%
%   f_sound is the frequency of the sound source, in Hz.
%   R_door is the reflection coefficient at the door (load) end
%   R_src is the reflection coefficient at the source wall
%   T_air is air temperature (use 297 if none given) in Kelvin.
%
if nargin > 4
    error('Too many inputs.')
end;
if nargin <= 3
    T_air = 297;    %degrees Kelvin
end;
if nargin <= 2
    R_src = 0.95;    %reflection coefficient at source wall
end;
if nargin ==1
    R_door = 0.99;    %reflection coefficient at the door
end;
if nargin <1
    error('No input arguments!  Can''t compute.')
end;

%definition of variables
R_air = 2.869E2;          %Gas constant, (J/kg/K)
k_air = 1.40;             %specific heat ratio
c_air = sqrt(k_air*R_air*T_air); %Eq. 1.20 in Munson
omega = 2*pi*f_sound;     %omega = 2*pi*f
beta = omega/c_air;       %page 415 in Rao
l_chamber = 8.79;         %8.79 m is the actual measurement
dist_rate = 0.1;          %the number of meters per sample
z = 0:dist_rate:l_chamber; %an array in .1 m steps for distance
L = length(z);            %L = # of samples along chamber
t_run = .5;               %length of time (in seconds) to run the simulation
sampling_rate = c_air/dist_rate; %number of samples per second
t = 1/sampling_rate:1/sampling_rate:t_run; %0.25 seconds is the length
                                         %of time you'd like to simulate the chamber

P_src = cos(omega.*t);    %driving function (source pressure)
Pp = zeros(length(t),length(z)); %allocate memory for positive-
```

```

Pn = zeros(length(t),length(z)); %going pressure wave
                                   %allocate memory for negative-
                                   %going pressure wave

%load the first sample
Pp(1) = P_src(1);
n = 1; %clock pulse

while n <= t_run*sampling_rate-1 %for each increment in time
    endpoint = Pp(n,L); %save endpoint
    begpoint = Pn(n,1); %save beginning point
    Pn(n,L) = endpoint*R_door; %load Pn(L)
    Pp(n,1) = begpoint*R_src+P_src(n); %load Pp(1)
    Pp(n+1,2:L)=Pp(n,1:L-1); %shift Pp(x) right
    Pn(n+1,1:L-1)=Pn(n,2:L); %shift Pm(x) left
    n = n + 1; %increment clock
    if mod(n,10) == 0 %every 10 steps, make a plot
        subplot(3,1,1),plot(z,Pp(n,:))
        axis([0 1_chamber -3 3])
        subplot(3,1,2),plot(z,Pn(n,:))
        axis([0 1_chamber -3 3])
        subplot(3,1,3),plot(z,Pp(n,:) + Pn(n,:))
        axis([0 1_chamber -5 5])
        xlabel('Distance (m)')
        drawnow
        M(n-1) = getframe;
    end;
end;

P_tot = Pp + Pn; %Total pressure standing wave is
                 %the superposition of the positive-
                 %going wave and the negative-
                 %going wave.

p = max(P_tot(:,1)); %maximum pressure achieved
return

```

## APPENDIX B

### CODE TO FIND $Q$ OF THE CHAMBER

The following MATLAB code was written to obtain results shown in Chapter 4.

Subroutines are also presented after the main program.

#### B.1 White Noise $Q$

```
function [frequencies,Qs] = white_noise_Q(folder)
%WHITE_NOISE_Q Calculates Qs of microphones in the chamber at CERL.
%
% [frequencies,Qs]=WHITE_NOISE_Q(FOLDER) makes use of data
% taken on the Yokogawa ScopeCorder and prepared by code
% written by Tim Eggerding at CERL. The folders that hold
% the shots must be renamed "01", "02", etc. Said folders
% must be in a main folder identified in the character
% string FOLDER. Valid for 99 shots or less only.
%
% There can be between 1 and 16 channels of data taken for
% any number of shots (or events). WHITE_NOISE_Q goes through
% FOLDER and averages the FFT of all shots.
%
% The averaged FFT is then smoothed by bin averaging and the
% Q is taken using the "3 dB down" bandwidth method.
%
% The frequencies at which resonant peaks are found are also
% saved in FREQUENCIES.
%
% See also YOKOGAWADATAGUI.
%
% R. Lee, 2004
% 10/01/04

load shotTimes %get the time of all of the shots
num_shots = length(shotTimeArray); %total number of shots to process
H_tot = 0; %initialize memory
bin_size = 22; %choose a bin size for bin averaging
%this parameter can be changed by the user
%at a later point in time based on the
%sampling rate and length of time of a
%single recorded shot

for k = 1:num_shots
%find out the name of the next folder
if k < 10 %if it's a single-digit number, like "1", make
%the folder name a 2-digit number like "01".
fol_name = sprintf('%s%s','0',num2str(k));
else %else if it's between 10-99, just name the folder
%the name of that number.
fol_name = num2str(k);
end;
cd(fol_name); %get into that individual folder

%get sampling rate; it's the same for all channels on the Yokogawa
cd CH1;load eventHeader;cd ..;clear traceName;clear date;clear time
%clear from memory parameters that aren't going to be used
```

```

num_channels = num_folders(cd); %the # of channels we're dealing
                                %with = the # of folders in the working directory

%put together data matrix by going into each channel's folder
%and getting the shot
data = build_data(num_channels);

H = abs(fft(data)); %take an FFT of the data matrix
d_f = samplingRate/size(H,1); %delta F, or the smallest change
                                %in frequency between points
freq_axis = 0:d_f:samplingRate-d_f;
                                %freq_axis is not used in future
                                %computation, but it is helpful
                                %to have it here for plotting
                                %purposes, should you want to
                                %view the FFT.

H_av = bin_average(H,bin_size);
save H H %save variable for later use
save H_av H_av
H_tot = H_tot + H_av; %add up the spectra
cd .. %return to parent directory
end;

H_tot = H_tot/num_shots; %average 'em out

%make a window
l_window = round(2*samplingRate); %length in # of samples
cutoff_freq = 100; %cutoff, in Hz, above which we
                    %don't care about peaks
num_iterate = round(cutoff_freq/d_f)-l_window+1; %# of times our
                    %window will have to be run through
maxx = max(H_tot(1:l_window,:)); %a list of maxima from each window
indicator = ones(1,num_channels); %initialize memory
                                %has range of [1:l_window]
locator = ones(1,num_channels); %locates a point
for k = 2:l_window
    maxx(k,:) = max(H_tot(k:k+l_window-1,:));
                                %the maximum value amongst the
                                %nearest samples
    for i_mic = 1:num_channels
        if maxx(k,i_mic) == maxx(k-1,i_mic) %if the maximum in this
            %set of samples is the same as the maximum of
            %the last set of samples (one window before)
            indicator(i_mic) = indicator(i_mic) + 1;%increment indicator
        else
            indicator(i_mic) = 1; %if there's a new maximum,
            %reset the indicator to 1.
        end;
    end;
end;

%at this point through (one run through with the window), the only
%possible local max could be at zero Hz (D.C.)
for i_mic = 1:num_channels
    if indicator(i_mic) == l_window %if so, |H(1,i_column)|
        local_max(i_mic) = H_tot(1,i_mic); %is a local max
        frequencies(i_mic) = 0; %zero for D.C.
        locator(i_mic) = locator(i_mic) + 1;%increment it
        indicator(i_mic) = 1; %reset indicator
    end;
end;

for i_mic = 1:num_channels %for each microphone
    for k = l_window+1:num_iterate %iterate through the FFT
        maxx(k,i_mic) = max(H_tot(k:k+l_window-1,i_mic));
    end;
end;

```

```

                                %max value in that window
if maxx(k,i_mic) == maxx(k-1,i_mic) %if this window's maxx is
    %the same as the last window's then increment the indicator
    indicator(i_mic) = indicator(i_mic) + 1;
else
    %else reset the indicator
    indicator(i_mic) = 1;
end;
if indicator(i_mic) == 1_window;
    %if so, then |H(1,i_column)| is a local max
    local_max(locator(i_mic),i_mic) = H_tot(k,i_mic);
    frequencies(locator(i_mic),i_mic) = freq_axis(k);
    %mark the frequency (Hz)
    locator(i_mic) = locator(i_mic) + 1;    %increment it
    indicator(i_mic) = 1;                    %reset indicator
    indices(locator(i_mic)-1,i_mic) = k;    %index in the FFT
end;
end;
end;

%bandwidth points are at 1/sqrt(2) times original peak
BW_point = local_max/sqrt(2);
value = local_max; %a copy of local_max so we can augment it

%get f_low for the "3 dB down" method
for i_mode = 1:size(frequencies,1)%do this for each resonant mode found
    for i_mic = 1:num_channels %do it for each channel
        offset = 0; %an offset (in samples) from f_0
        while value(i_mode,i_mic) > BW_point(i_mode,i_mic)
            value(i_mode,i_mic) = H_tot(indices(i_mode,i_mic)-(offset+1),i_mic);
            offset = offset + 1;
        %increment the offset as long as you're still above the 3 dB down point
        end;
        if indices(i_mode,i_mic)-offset >= 1
            %if the index were ever recorded,
            %do some interpolation to calculate a low frequency to mark
            %the bandwidth point for the "3 dB down" method"
            slope = H_tot(indices(i_mode,i_mic)-offset+1,i_mic) -
                    H_tot(indices(i_mode,i_mic)-offset,i_mic);
            rise = BW_point(i_mode,i_mic)-H_tot(indices(i_mode,i_mic)-offset,i_mic);
            f_low(i_mode,i_mic) = (rise/slope+indices(i_mode,i_mic)-offset)*d_f;
        end;
    end;
end;

%get f_high for the "3 dB down" method
value = local_max;
for i_mode = 1:size(frequencies,1)%do this for each resonant mode found
    for i_mic = 1:num_channels %do it for each channel
        offset = 0; %an offset (in samples) from f_0
        while value(i_mode,i_mic) > BW_point(i_mode,i_mic)
            value(i_mode,i_mic) = H_tot(indices(i_mode,i_mic)+(offset+1),i_mic);
            offset = offset + 1;
        %increment the offset as long as you're still above the 3 dB down point
        end;
        if indices(i_mode,i_mic)+offset-1 >= 1
            %if the index were ever recorded,
            %do some interpolation to calculate a high frequency to mark
            %the bandwidth point for the "3 dB down" method"
            slope = H_tot(indices(i_mode,i_mic)+offset,i_mic) -
                    H_tot(indices(i_mode,i_mic)+offset-1,i_mic);
            rise = BW_point(i_mode,i_mic)-H_tot(indices(i_mode,i_mic)+offset-1,i_mic);
            f_high(i_mode,i_mic) = (rise/slope+indices(i_mode,i_mic)+offset-1)*d_f;
        end;
    end;
end;

```

```

        end;
    end;
    Qs = frequencies./(f_high - f_low); %the definition of the 3 dB down Q
    return

```

## B.2 Number of Folders Subroutine

```

function [num_channels] = num_folders(folder)
%NUM_CHANNELS = NUM_FOLDERS(FOLDER)
% Num_folders finds the number of folders within a
% folder.
%
% 'folder' is any string that contains the location
% of the main folder
%
% Ryan Lee, 2004

return_path = cd; %save the current path
                %so that you can return to it later.
cd(folder);      %make 'folder' your current directory
d = dir;
num_channels = 0; %initialize # of channels to zero
for i = 3:size(d,1) %start at i = 3 because i = 1 is not for a folder,
                    %but rather the "." in DOS commands, and the i = 2
                    %is for the ".." DOS folder, which is definitely
                    %not of interest here.
    if d(i).isdir == 1 %if, out of all of the files and folders in
                        %our folder, we come across a directory (folder)
        num_channels = num_channels + 1; %add 1 to num_channels for
    end; %every folder encountered
end;
cd(return_path); %return to the original directory
return

```

## B.3 Build Data Subroutine

```

function [data] = build_data(num_channels)
%[DATA] = BUILD_DATA(NUM_CHANNELS)
% Puts together a matrix with data from various channels
% of the Yokagawa ScopeCorder in columns.
%
%Written for handling data from the Yokagawa ScopeCorder
%Version 1.0 completed 13 JUL 2004 by:
%Ryan Lee

parent_dir = cd;
%get the sensitivities of all the mics
sensitivities = build_sensitivities(num_channels);
d = dir;column = 1;

for i = 1:num_channels %for each of the channels
    cd(sprintf('%s\\%s%s',parent_dir,'CH',num2str(i)));%get into folder
    load eventWaveform.mat; %this waveform is in volts!
    data(:,column) = waveform*sensitivities(column);
    %this column of 'data' gets this folder's waveform
    %units will be in pascals
    if length(waveform) ~= size(data,1)
        error('Channel data is of different lengths!')
    end;
    clear waveform %save memory by clearing variable

```

```

        column = column + 1;
        cd(parent_dir);
end;
return

```

## B.4 Bin Average Subroutine

```

function [H_out]= bin_average(H,bin_size)
%[H_OUT]= BIN_AVERAGE(H,BIN_SIZE)
%   Bin_average takes any data vector or matrix and smoothes out the
%   curve by bin averaging.  The program takes the nearest number of
%   points (dictated by bin_size input variable) and averages them
%   to get a value for that particular point.
%
%   Hence, size(H_out) will equal size(H).
%
%   If H is 2-dimensional, data should be in the columns of H, not the
%   rows.
%
%   @Ryan Lee
%   08/13/04

bin_size = 2*round(bin_size/2); %make it an even number

H_out = zeros(size(H));           %initialize memory
H_out(1:bin_size/2,:) = H(1:bin_size/2,:);
for i_column = 1:size(H,2)        %for each signal in H
    for i = bin_size/2+1:size(H,1)-bin_size/2
        %RMS bin average

        H_out(i,i_column) =
            sqrt(mean(H(i-bin_size/2:i+bin_size/2,i_column).^2));
    end;
end;
return

```



## APPENDIX C

### CODE TO FIND BEST FIT STANDING WAVE

The following MATLAB code was written to obtain results shown in Chapter 6.

#### C.1 Standing Wave Fitter Routine

```
function [Pi,gamma,theta,SWR] = sw_fitter(filename,locations)
%SW_FITTER Calculates parameters of a transmission line terminated
% by a resistive load as an analogy to a standing wave in the
% chamber at US Army CERL. 13 microphone ports on the side of
% the chamber must be used unless the user manually changes
% the variable "loc" on line __.
%
% The chamber can be empty, which results in a line terminated
% by a short circuit, or may have a material placed inside
% the chamber door. For current thesis research, the material
% of choice is vegetation mass.
%
% Parameters output are Pi, the magnitude of an incident
% pressure wave, gamma, the magnitude of the reflection
% coefficient at the chamber door due to the added impedance;
% theta, the phase angle of the reflection coefficient (gamma
% will be complex); and SWR, the standing wave ratio.
%
% [Pi,gamma,theta,SWR]=SW_FITTER(FILENAME,LOCATION)
%

%% DEFINE CONSTANTS
location_rate = 1000;           %samples per meter
l_chamber = 8.798;              %length of chamber in meters
chamber_axis = 0:l/location_rate:l_chamber;
T_air = 297;                    %air temp in Kelvin
R_air = 2.869E2;                %Gas constant, (J/kg/K)
k_air = 1.40;                   %specific heat ratio
mm_HG = 762;                    %mm of HG for barometric pressure; 762 mm = 30 in.
gamma_HG = 133;                 %specific weight of mercury (kN/m^3)
p_atm = gamma_HG*mm_HG;         %atmospheric pressure
rho_air = p_atm/(R_air*T_air);  %kg/m^3
c_air = sqrt(k_air*R_air*T_air); %Eq. 1.20 in Munson
Z_0 = rho_air*c_air/4;          %Z = rho*c/A where A = 4 m^2
gamma_R = 0.99;                 %reflection coefficient at door of chamber
%get sampling rate
cd CH1;load eventHeader;cd ..;clear traceName;clear date;clear time

loc=[0.744 1.354 1.964 2.574 3.184 3.794 4.404
      5.014 5.624 6.234 6.844 7.454 8.064]';
%locations of microphones

%% OPEN THE FILES AND READ THEM IN
num_channels = num_folders(cd); %the # of channels we're dealing with =
% # of folders in the working directory

%put together data matrix by going into each
%channel's folder and getting the shot
data = build_data(num_channels);

%% FIND THE FREQUENCY OF THE EXCITATION
[amplitude,freq]=freq_finder(data,samplingRate);
```

```

%outputs an array of peak-peak amplitudes and frequencies
time_axis = 1/samplingRate:1/samplingRate:size(data,1)/samplingRate;
beta = 2*pi*freq/c_air; %in radians

%%% MAKE SOME INITIAL GUESSES
%tabulate all of the various amplitudes
amp = sort(amplitude(:),'descend');
%make the largest mic's amplitude your initial guess
abs_Pi = amp(1);clear amp;
%an incremental amount to differentiate guesses of theta
d_theta = pi/50;
%initial array of a bunch of guesses for theta
theta = d_theta:d_theta:2*pi;
%an incremental amount to differentiate guesses of Pi
d_Pi = abs_Pi/50;
%initial array of a bunch of guesses for Pi
Pi = 0.5*abs_Pi:d_Pi:(3/2)*abs_Pi-d_Pi;
%an incremental amount to differentiate guesses of gamma
d_gamma = 0.02;
%initial array of a bunch of guesses for magnitude of gamma
gamma = d_gamma:d_gamma:1;

%the following 3 for loops generate a 4-dimensional matrix
%consisting of all of the combinations of gamma, theta,
%mic location, and Pi amplitude you can get
for i_gamma = 1:length(gamma)
    for i_Pi = 1:length(Pi)
        for i_theta = 1:length(theta)
            p(i_theta,:,i_gamma,i_Pi) =
                Pi(i_Pi)*sqrt(1+gamma(i_gamma)^2+2*gamma(i_gamma)*cos(
                    2*beta*loc+theta(i_theta)))';
        end;
    end;
end;

%Create an error matrix that's 1 dimension less than the dimension of p
for i_gamma = 1:length(gamma)
    for i_Pi = 1:length(Pi)
        for i_theta = 1:length(theta)
            for i_loc = 1:length(loc)
                e(i_theta,i_loc,i_gamma,i_Pi) = (amplitude(i_loc) -
                    p(i_theta,i_loc,i_gamma,i_Pi))^2;
            end;
        end;
    end;
end;

error = sum(e,2);clear e; clear p %summing 13 individual errors
%throw away the actual value, but keep the index
[throw_away,theta_index] = min(min(min(error,[],4),[],3));
%throw away the actual value, but keep the index
[throw_away,Pi_index] = min(min(min(error,[],3)));
%throw away the actual value, but keep the index
[throw_away,gamma_index] = min(min(min(error,[],4)));

theta = theta(theta_index); %LMS result for theta
Pi = Pi(Pi_index); %LMS result for |Pi| amplitude
gamma = gamma(gamma_index); %LMS result for |gamma|
p = Pi*sqrt(1+gamma^2+2*gamma*cos(2*beta*chamber_axis+theta));
SWR = max(p)/min(p); %standing wave ratio
%define complex reflection coefficient
clear i;gamma_i = gamma*exp(i*theta);
Z_R = rho_air*c_air*(1+gamma_i)/(1-gamma_i); %kg/m^2/s
plot(chamber_axis,Pi*sqrt(1+gamma^2+2*gamma*

```

```

                                cos(2*beta*chamber_axis+theta)))
hold;plot(loc,amplitude,'ro');xlabel('Chamber axis (m)')
title(sprintf('%s%0.3g%s%0.3g%s%0.3g%s%0.3g%s%0.3g%s', 'f = ',
              freq, ' Hz, Gamma = ', gamma, ', SWR = ', SWR,
              ', Z_L = ', real(Z_R), ' + ', imag(Z_R), 'i.'))
return

```

## C.2 Frequency Finder Subroutine

```

function [amplitude,frequency]=freq_finder(data,sampling_rate)
%FREQ_FINDER finds the main frequencies of waveforms
% and also the amplitudes of those signals for
% that particular frequency. The signal can be a cal
% signal or any other signal with a main resonance that
% can clearly be seen in the form of a spike
% in the FFT of the data that is in 'data'.
% That main resonance must be above 16 Hz.
%
% The data must be set up in columns (as opposed
% to rows) in the data input matrix, and there
% can be more than 1 waveform.
%
% [amplitude,omega]=freq_finder(data,sampling_rate)
%

for i_column = 1:size(data,2) %for each waveform remove DC offset
    data(:,i_column) = data(:,i_column)-mean(data(:,i_column));
end;

fft_length = 2^(nextpow2(size(data,1))-1); %this will be the length
                                         %of the FFT and the "-1" insures that our data will be
                                         %truncated (rather than zero padded) in the FFT algorithm
freq_res = sampling_rate/fft_length; %frequency resolution (Hz)

% FIND THE FREQUENCY OF THE SIGNALS
num_sigs = size(data,2); %the # of signals we're dealing with
H = fft(data,fft_length); %takes the FFT of the columns
% IMPORTANT NOTE: H(1) is really the element of the
% FFT that is at a frequency of 0. Matlab does not
% index any matrices beginning with 0, it always
% begins with an index of 1. Therefore H(0) is
% improper in Matlab.
% freq_axis = 0:freq_res:sampling_rate-freq_res;
%the location (an index) of 16 Hz on the frequency axis
pointer = (round(16/freq_res)+1)*ones(1,num_sigs);
if pointer(1) < fft_length
    H(1:pointer(1),:) = 0; %zero out everything below 16 Hz
    %zero out everything above -16 Hz
    H(fft_length+1-pointer(1):fft_length,:) = 0;
else
    error('FFT is not long enough or Sampling rate too small.');
```

data was in rows instead of columns.');

```

end;

for i_sig = 1:num_sigs %for each waveform
    %keep going as long as max hasn't been reached
    while abs(H(pointer(i_sig),i_sig)) < max(abs(H(:,i_sig)))
        %increment pointer along the freq. axis of the FFT(data)
        pointer(i_sig) = pointer(i_sig) + 1;
    end;
    %in Hz, get frequency of that point that's max in the FFT
    freq(i_sig) = pointer(i_sig)*freq_res-freq_res;

```

```

end;

frequency = mode(freq);           %finds the mode of these frequencies
                                   %and eliminates outliers

%some cutoff frequencies
%signal freq - a semitone
cutoff_1 = round(frequency*2^(-1/12)/freq_res)+1;
%signal freq + a semitone
cutoff_2 = round(frequency*2^(1/12)/freq_res)+1;
cutoff_3 = size(H,1)+2 - cutoff_2; %the negative frequency of freq_2
cutoff_4 = size(H,1)+2 - cutoff_1; %the negative frequency of freq_1
block_size = round(sampling_rate/frequency); %in units of samples/cycle

%an easy way to zero out all unwanted low and high frequencies from
%zero to one semitone below our calibration frequency
H(1:cutoff_1,:) = 0;

%from one semitone above our calibration freq. to one semitone below
H(cutoff_2:cutoff_3,:) = 0; %the negative cal signal frequency

%from one semitone above the negative cal signal frequency to
H(cutoff_4:length(H),:) = 0; %the end of the FFT zero it out.

idata = real(ifft(H,size(H,1))); %a new time-based signal. This
%should be similar to the original signal minus any unwanted noise.
%Its mean should be zero

for i_sig = 1:num_sigs %for each waveform
% FIND AMPLITUDES OF THE SIGNALS
    n=1;
    %block data up and find max and mins
    for num_block = 1:floor(fft_length/block_size) %for each block
        pos_pt(num_block) = max(idata((num_block-
            1)*block_size+1:num_block*block_size,i_sig));
        neg_pt(num_block) = min(idata((num_block-
            1)*block_size+1:num_block*block_size,i_sig));
    end;
    amplitude(i_sig) = (mean(pos_pt)-mean(neg_pt))/2;
end;
return

```

## REFERENCES

- <sup>1</sup>D. I. Cook and D. F. Van Haverbeke, "Trees and shrubs for noise abatement," University of Nebraska, U.S. Department of Agriculture Research Bulletin 246, 1971.
- <sup>2</sup>S. H. Burns, "The absorption of sound by pine trees," in *The Journal of the Acoustical Society of America* **65** (3), 658-661 (1979).
- <sup>3</sup>G. W. Swenson, Jr., "A standing-wave facility for low-frequency impedance/absorption measurement," *Applied Acoustics* **40**, 355-362 (1993).
- <sup>4</sup>G. W. Swenson, Jr., *Principles of Modern Acoustics* (D. Van Nostrand Company, Inc, New York, 1953), pp. 49, 90, 153-157, 172.
- <sup>5</sup>L. E. Kinsler, A. R. Frey, A. B. Coopers, and J. V. Sanders, *Fundamentals of Acoustics*, 3<sup>rd</sup> ed. (John Wiley & Sons, Inc., New York, 1982), pp. 32-33, 107, 208, 231, 233.
- <sup>6</sup>B. R. Munson, D. F. Young, and T. H. Okiishi, *Fundamentals of Fluid Mechanics*, 2<sup>nd</sup> ed. (John Wiley & Sons, Inc., New York, 1994), pp. 15, 27, 46.
- <sup>7</sup>B. Gold and N. Morgan, *Speech and Audio Signal Processing* (John Wiley & Sons, Inc., New York, 2000), pp. 141-143, 176.
- <sup>8</sup>D. R. Glover, NASA's Glenn Research Center, "Dictionary of technical terms for aerospace use," 1965, <http://roland.lerc.nasa.gov/~dglover/dictionary/q.html>.
- <sup>9</sup>R. C. Dorf and J. A. Svoboda, *Introduction to Electric Circuits*, 5<sup>th</sup> ed. (John Wiley & Sons, Inc., New York, 2001), pp. 602-605.
- <sup>10</sup>V. Dickason, *The Loudspeaker Design Cookbook*, 6<sup>th</sup> ed. (Audio Amateur Press Publishers, Peterborough, New Hampshire, 2000), p. 111.
- <sup>11</sup>N. N. Rao, *Elements of Engineering Electromagnetics*, 5<sup>th</sup> ed. (Prentice Hall, Upper Saddle River, New Jersey, 2000), pp. 250-251, 414, 419, 420, 430, 611.
- <sup>12</sup>F. V. Hunt, *Electroacoustics, The Analysis of Transduction, and Its Historical Background* (American Institute of Physics for the Acoustical Society of America, Melville, NY, 1982), p. 99.
- <sup>13</sup>J. L. Herring, "Developments in transmission-line modeling method for electromagnetic compatibility studies," Ph.D. dissertation, University of Nottingham, United Kingdom, 1993, pp. 127-132.
- <sup>14</sup>L. L. Beranek, *Acoustics* (Acoustical Society of America through the American Institute of Physics, Inc., New York, 1996), p. 24, 299, 301.
- <sup>15</sup>J. A. Kong, *Electromagnetic Wave Theory* (EMW Publishing, Cambridge, Massachusetts, 2000), pp. 183, 191, 206, 208-209.

<sup>16</sup>J. Lee and G. W. Swenson, Jr., "Operation manual for the low-frequency acoustic impedance measurement facilities," USA CERL, Champaign, IL, (1993).

